Игры с чипом
Цикл статей для обучения детей основам информатики
ведёт доктор физико-математических наук А.А.Мигдал

SHOUT
SHARE
IT

Annotation

Цикл детских образовательных статей из журнала "Пионер" за 1986-1987 года.

В сказочно-игровой форме для дошкольников и младших школьников даются базовые понятия информатики.

Предисловие для ребят и родителей

Мы приближаемся к новому веку, в котором люди самых разных профессий будут работать на компьютерах — электронно-вычислительных машинах.

Как человек должен излагать свои мысли, чтобы его понял компьютер? А как компьютер будет понимать человека? Эти и многие другие интересные задачи ставит информатика.

Их придется решать тем, кто сейчас учится в школе, и тем, кто только ходит в детский сад, и тем, кто еще не родился на свет. Им надо научиться мыслить более точно, работать более организованно, чем это удается нам, старшему поколению.

В этом году впервые начато обучение школьников основам информатики. Этот предмет преподают в 9—10-х классах, а младшим школьникам пока что удается в лучшем случае поиграть с электронной игрушкой или с калькулятором. Поэтому они видят лишь внешнюю сторону дела: дисплей на жидких кристаллах, кнопки, мелькающие цифры... Но как передать им важные идеи информатики?

Мы думаем — в игровой форме, на примере сказок, стихов и головоломок.

Интересно ли это будет вам, ребята? Понравится ли вам новый сказочный персонаж — веселый, смышленый и задиристый Чип из калькулятора? Ответьте «Пионеру» на эти вопросы.

Мы рассчитываем и на вашу помощь, дорогие родители, и ждем от вас и ваших детей откликов на этот раздел и полезных предложений.

И еще — последнее. Товарищи родители, не прячьте от ребят калькулятор: эта игрушка гораздо нужнее им, чем вам. И не бойтесь, что они разучатся считать. Калькулятор, особенно программируемый, учит не только считать, он учит думать.

- Игры с Чипом
  - Чип из калькулятора
  - Салочки-считалочки
  - Салочки с прыжками
  - Сказки-программы

- Царевич выбирает невесту
- Как спасти колобка
- Случай в квартире 130
- Новая игра
- Барон Мюнхгаузен и урок физкультуры
- Конкурс поросят
- Электронная яблоня
- Как уговорить маму купить жирафа
- Чип и Сережа читают ваши письма
- Введение
- Сказки восточного базара
- Роза для королевы
- Литературный экскурс в историю счетных машин
- Заклинание для лентяев
- Приключения электронного мальчика с пальчик
- Чип и Сережа называют победителей
- Как угадать день рождения друга и вывести братьев из Болота Ошибок
- Чем кончились мучения Евклида
- Двадцать спичек и монета
- Совет Чеширского кота
- Случайные числа
- До свиданья, друзья!
- Как обхитрить Мегафлопа
- Двоичный поиск и влюбленный принц
- Бурная жизнь Фатландии
- Чип и Сережа читают ваши письма
- В поисках выхода
- Секретное послание
- Рекурсивный крокет
- В ожидании Чипа
- Об электронных переводчиках и лучших портных Кливленда
- Работа над ошибками

- notes
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
- comments
  - 1

○ 2

Игры с Чипом
ведёт доктор физико-
математических наук А.А.Мигдал

Чип из калькулятора

Сережа сидел за столом, уныло глядя в тетрадь по математике. Время от времени он вздыхал и чесал за ухом шариковой ручкой. Тщетно! Задача не решалась, и помощи ждать было неоткуда.

Сережа открыл ящик стола и достал оттуда маленький красивый калькулятор, который ему подарила бабушка на день рождения. Вообще-то в школе не разрешали делать уроки на калькуляторе: «Считать разучитесь». «Ну ладно, — решил Сережа, — я чуть-чуть на калькуляторе попробую, а если получится, я потом сам пересчитаю, честное слово».

Но и это не помогло. Сережа нажимал все цифры, складывал, вычитал и умножал, но все равно задача не решалась. «Вот дурацкая машина,— разозлился он на калькулятор, — только и знает, что цифрами мигать, а задачу решить не может».

И тут Сереже пришла в голову заманчивая мысль. Как было бы интересно разобрать калькулятор — не до конца, а так, чтобы он все еще работал, — и посмотреть, что же там внутри происходит, когда он цифры складывает. Что-то же должно происходить? Конечно, он не бормочет, как человек, «один пишем, два в уме... э... э... э, сколько же это будет?» Но тогда как же он считает? На пальцах, что ли? Интересно, что там у него вместо пальцев и как он их загибает?

После недолгой борьбы любопытство пересилило страх, и Сережа начал потихоньку отворачивать винтики. Вот они уже лежат на столе, осталось только снять крышку. Да, но как же он подглядит, как калькулятор считает?

Ага, надо перед тем, как крышку приоткрыть, заставить его какие-нибудь цифры складывать. Какие же? Ну, что-нибудь посложнее, например: 1234 прибавить к 5678. Так... Готово!

Сережа нажал на кнопку «+», осторожно приоткрыл крышку, и... из калькулятора выпрыгнуло невиданное существо! Не больше жука, с головой, ручками и ножками, как человечек, только туловище не круглое, а квадратное и плоское. С обоих боков блестящие выступы — маленькие лапки. Глазки так и сверкают, а на голове серебристая шапочка с таинственными знаками. Выражение личика заносчивое и гордое: мол, я не букашка, так что полегче, дружок.

Существо уселось на учебник по математике, закинуло ножку на ножку и с удовлетворением пискнуло: « У...уф— наконец-то!»

Сережа протер глаза. Существо не исчезло, наоборот, оно подмигнуло ему и хихикнуло.

— Что, глазам не веришь? Сам вызывает, а потом вон как встречает. Смотри, я могу обидеться и уйти.

— Ты... ты кто такой? — спросил ошарашенный мальчик.

Квадратный человечек спрыгнул с учебника, сорвал с головы шапочку и лихо шаркнул ножкой.

— Меня зовут Чип, а работаю я вот там, — Чип показал на разобранный калькулятор [1]. — Скучно... Складывай да умножай, вычитай да дели. Вот у больших компьютеров работа интересная: они и ракеты водят, и пароходы, и стихи сочиняют, и в шахматы играют, и даже программы для других компьютеров пишут.

— Так ты что... ты и есть калькулятор? Как же ты считаешь, пальцы загибаешь, что ли?

Чип снова уселся на учебник, поджал ножки под себя и с важным видом поднял палец вверх.

— Пальцы загибать тоже с умом надо. Вот ты, например, до скольких можешь на пальцах сосчитать?

— Как до скольких? — удивился Сережа. — До десяти, конечно. Это — если без ног.

— Вот как, без ног только до десяти. А я без ног до тысячи могу сосчитать, а с ногами и до миллиона. И тебя могу научить — это так просто. Смотри, — он сжал правый кулачок, — когда ни одного пальца — это, конечно, ноль, — он поднял мизинец, — так 1, а дальше мы мизинец опускаем, а безымянный палец поднимаем — это 2.

— Как же 2, когда палец-то один, — вмешался Сережа.

— Это по-вашему один, а я не пальцы считаю, а числа. Есть же разница между мизинцем и безымянным пальцем? Значит, можно мизинцем обозначать 1, а безымянным 2. А вот теперь мы снова мизинец поднимаем. Получается два пальца — безымянный и мизинец. По-вашему это 2, а по-моему — 3. Почему 3? Очень просто. Мизинец — 1, безымянный — 2, вместе — 3. Теперь эти

два опускаем, средний палец поднимаем. Получается — 4.

— Дальше снова начинаем с мизинца, — сказал Сережа, который кое-что начал соображать. — Средний и мизинец — 5, средний и безымянный — 6, средний, безымянный и мизинец — 7, а указательный — 8.

— Правильно. А ты заметил, что каждый новый палец в два раза больше предыдущего стоит? Так и дальше будет. Значит, указательный — 8, большой — 16, тут мы вторую руку приставляем, левый большой палец — 32, левый указательный — 64, левый средний — 128, левый безымянный — 256 и левый мизинец — 512.

— Ага, — засмеялся Сережа, — все-таки до пятисот, а не до тысячи, как ты хвастался.

Маленький человечек надулся от обиды:

— Прошу запомнить раз и навсегда, калькуляторы, если они в своем уме, конечно, никогда не врут и не хвастаются. Слава богу, этому нас люди не научили. Смотри дальше: это левый мизинец — 512, а если прибавить левый безымянный, то будет 512 + 256 = 768. Теперь поднимаем еще левый средний, будет 768 + 128 = 896. Мало? Ну тогда поднимем еще левый указательный, будет 896 + 64 = 960. Что, еще надо? Тогда поднимем левый большой: 960 + 32 =992. Ну, а сколько на правой руке, мы уже знаем 16 + 8 + 4 + 2 + 1 = 31. Итого — 1023, как и обещал, даже больше. Сдачи не надо!

Чип гордо поднял голову в серебряной шапочке. Сережа зааплодировал. Он уже понял, что Чип очень любит, когда им восхищаются.

— Чип, какой ты умный! А скажи, как это тебе удается все эти цифры запомнить?

Чип так и расцвел. Он уже открыл рот, чтобы сказать, но потом подозрительно посмотрел на Сережу.

— А смеяться не будешь?

— Ну что ты!

— Видишь ли, главное — запомнить степени двойки: 2, 4, 8 и т. д. Каждый запоминает, как может, а у меня для этого считалка есть. Вообще-то меня не запрограммировали стихи сочинять, это я сам в свободное от работы время. — Чип густо покраснел. — Будешь смеяться — обижусь, уйду и никогда не приду.

— Честное-пречестное слово, чтоб мне провалиться на этом месте, если засмеюсь.

— Ну, уж ладно.

Слон живет у нас в квартире,
В доме **два**, подъезд **четыре**.
По часам привык питаться:
Утром в **восемь**, днем в **шестнадцать**.
Ест на завтрак непременно
Тридцать две охапки сена.
После утренней прогулки —
Шестьдесят четыре булки.
На обед ему приносим
Огурцов **сто двадцать восемь**.
Помидоров может съесть
Двести и пятьдесят шесть.
Съест блинов **пятьсот двенадцать**,
Это если не стараться.
А замесишь на кефире —
Тысячу двадцать четыре.

Сережа изо всех сил закусил губу, чтобы не рассмеяться.

— Замечательные стихи, — пробормотал он сдавленным от смеха голосом. — А главное, так легко запомнить: «Ест на завтрак непременно тридцать две охапки сена». Теперь уж я не забуду степени двойки.

— Кое-что соображаешь, — пропищал Чип, — хоть и жалуешься, что задачу решить не можешь. Кстати, в той задаче не надо ли сначала поинтересоваться, сколько стоит бутылка без пробки, но с кефиром? Это, правда, не моего ума дело, мое дело цифры складывать, какие прикажут. Ну да ладно, ты подумай, а мне пора на работу: у меня там все регистры простаивают.

— Подожди, какие регистры? Не уходи, — взмолился Сережа, — с тобой так интересно.

— В следующий раз, — Чип улыбнулся. — Захочешь вызвать, сложи 1234 и 5678, а захочешь обратно меня вернуть в калькулятор — вычти. Ну, давай, а то уже поздно. До скорого свидания!

Через десять минут Сережа был уже в кровати. В столе лежал собранный калькулятор, а в портфеле — тетрадь по математике с решенной задачей.

Салочки-считалочки

На следующий день Сережа с трудом дотерпел до вечера, так ему хотелось встретиться с Чипом. Когда наконец уроки были сделаны, Сережа закрылся в своей комнате, вынул из стола калькулятор и набрал заклинание: 1234 + 5678.

— 6912, — весело крикнул Чип, выпрыгивая из калькулятора . — Привет! Ну, чем сегодня займемся?

— В прошлый раз ты говорил, что до миллиона можешь на пальцах считать. Это странно. Если на руках тысяча, то на ногах тоже тысяча, а вместе — две тысячи? Постой-постой... Ты, наверное, опять будешь по одному пальцу приставлять и все удваивать. Это значит, при каждой комбинации пальцев ног и при каждой комбинации пальцев рук ты будешь новое число получать? Все понял: надо числа комбинаций на ногах, то есть 1023, **перемножить** на такие же числа комбинаций на руках. Ну, для ровного счета пусть 1000 на 1000 — это и будет миллион!.. А расскажи мне, как ты складываешь, — попросил Сережа.

— А я вовсе и не складываю. У меня пальцы в салочки-считалочки играют.

— Что еще за салочки-считалочки? Я таких салочек не знаю.

— А вот какие салочки. — Чип сжал кулачки и приставил их пальцами друг к другу, так что каждый палец против одноименного оказался.

— Левая рука — салочка, — продолжал Чип, — смотри, я левым мизинцем осалил правый. А правила игры такие: если ты сидел, когда тебя осалили, — ты встаешь, а если уже стоял, ты левого соседа салишь и садишься. Вот и все!

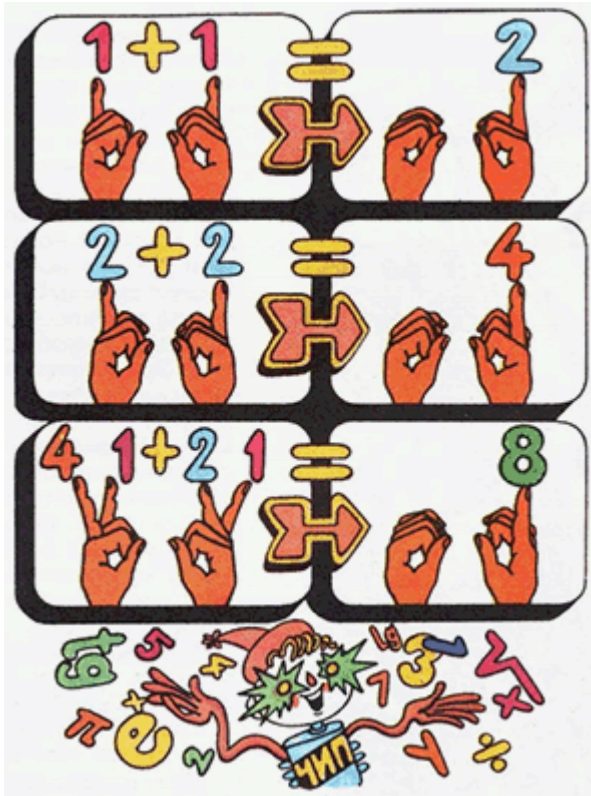
— Ну и при чем же тут сложение? — удивился Сережа.

— А это и есть сложение. — Чип хитро прищурился. — Разве не видишь? Давай сложи 1 и 1. Один уже есть — мизинец на правой руке. Вот я осалил его левым мизинцем, он сел, а правый безымянный встал. Помнишь, сколько это? Два. Значит, один и один — два. А теперь левым безымянным правый осалим: тот уже стоит, значит, ему надо средний правый осалить, а самому сесть. Средний правый встал. Это сколько? Четыре. Два и два — 4.

— Э...э, — протянул Сережа, — это ты нарочно так подбираешь. А вот сложи-ка 5 и 3 с помощью своих салочек.

— Пожалуйста, — с готовностью ответил Чип. — Значит, 5 — это мизинец да средний палец, поднимаем их на левой руке. Три — это мизинец да безымянный палец, поднимаем их на правой. Левым мизинцем осалим правый. Тот осалит правый безымянный, а сам сядет. Правый безымянный осалит правый средний, сам

сядет, а правый средний встанет. Теперь осталось левым средним осалить правый средний. Тот осалит правый указательный, а сам сядет. Что получилось? Указательный палец — это 8, значит, 5 + 3 = 8. Видишь, все правильно получается, только надо по очереди салить левыми пальцами такие же напротив, а те уже сами знают, что делать.



— И что же, так можно до тысячи складывать? — недоверчиво спросил Сережа.

— Ну, на руках — нет, пальцев не хватит, — ответил Чип. — Но ведь это я тебе на пальцах объясняю, а там, — он показал на калькулятор, — у меня целое хозяйство вместо пальцев, там у меня биты. Знаешь, как они быстро в салочки-считалочки играют? Ого-го.

— Послушай, а как же ты умножаешь?

— Не все сразу, — ответил Чип и подмигнул Сереже. — Много будешь знать — скоро состаришься. Это мы оставим на следующий раз.

Салочки с прыжками

На следующий день Сережа сделал уроки, вызвал Чипа, и сначала они просто поболтали о том и о сем.

Когда они вдоволь посмеялись и поиграли, Сережа спросил его:

— Чип, а как же ты все-таки умножаешь? В прошлый раз ты так и не успел рассказать.

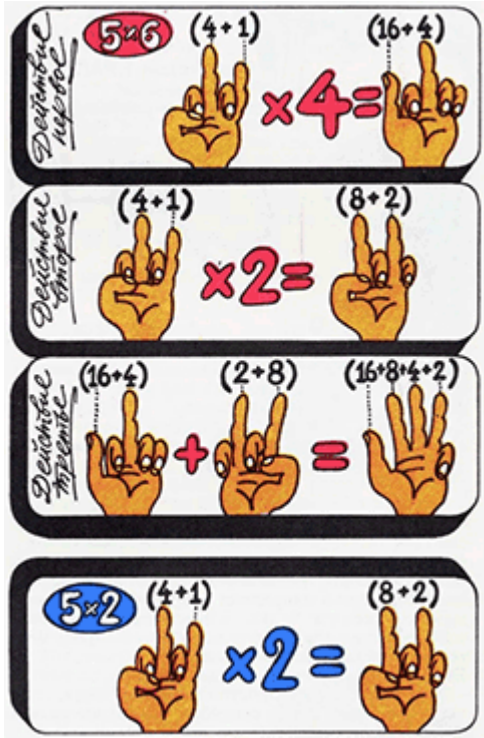
— А ты хорошо усвоил, как я складываю?

— Да, ты же меня научил — салочки пальцами.

— Ну, а здесь те же салочки, только с прыжками. Например, хочешь ты число на 2 умножить — сдвинь все пальцы на один, хочешь на 4 умножить — сдвинь на 2 и т д. Скажем, было у тебя число 5 — это 4+1, то есть мизинец и средний палец — помнишь? Умножим его на 2: 4 превратится в 8, а 1 превратится в 2. Получится безымянный палец и указательный. На один палец прыгнули влево. Ну, а если 5 умножить на 4, то все прыгнет на 2 влево, получится большой палец и средний — 20.

— Так это прыжки, — сказал Сережа,— а где же салочки?

— А салочки будут, если ты не на степень двойки (то есть не на 2, 4, 8 и т. д.) хочешь умножать, а на какое-нибудь другое число, например, на 6. Вот посмотри: 6 - это 4+2, значит, надо умножить наше число 5 сначала на 4 (прыгнуть на два шага влево), а потом на 2 (прыгнуть на 1 шаг влево) и сложить. 5x6=5x4+5x2=20+10=30. Ну, а складываем мы, как вчера: с помощью салочек. Так и получаются салочки с прыжками. Прыжок — салочки, прыжок — салочки. А если бы мы те же 5 умножили, например, на 7? Это 4+2+1, то есть мизинец, безымянный и средний пальцы. Будет прыжок на 1 шаг (когда 5 на 2 умножили), потом салочки, когда с исходным числом 5 сложили. Получится 15, а потом прыжок на 2 шага, когда 5 на 4 умножили и получили 20, потом снова салочки, когда к предыдущему числу 15 прибавили и получили 35. Вот тебе и все умножение.



— Чип, — задумчиво сказал Сережа, — а вот скажи, ты что, никогда не устаешь? Никогда тебе не надоедает все правильно делать? Пошалить никогда не хочется?

— Как не хочется? Иногда еще как хочется пошалить и нарочно напутать, — ответил Чип. — Только ни один компьютер, ни один калькулятор еще ни разу нарочно никому не навредил. Нечаянно бывает, если что-нибудь сломается или если неправильную задачу дадут, а нарочно — никогда. Никогда! Ну ладно, мне пора домой, а то мы заболтались.

Сказки-программы

На следующий день, когда Сережа перед сном вызвал Чипа и они немножко поиграли в салочки-считалочки, Сережа попросил:

— Чип, а ты сказки какие-нибудь знаешь? Я так люблю сказки, даже совсем глупые, для малышей.

Чип задумался.

— Сказку? Нет, это работа для суперкомпьютера, вот тот может хоть сто тысяч сказок в секунду рассказывать.

— Да куда мне столько, мне и одной хватит. Ну, пожалуйста, Чипка, я очень тебя прошу. Попробуй, может, что-нибудь получится. Ты такой умный!

— Что ж, — Чип был явно польщен, — сказку, говоришь? Ладно, попробуем, только это будет не сказка, а сказка-программа. Ручка есть? Записывай.

И вот что продиктовал Чип.

РЕПКА. *Сказка-программа для детей младшего и среднего школьного возраста, особенно для тех, кто занимается в кружке юных программистов.*

```
Глава № 1. Жили-были:
жилец № 1 = Дедка,
жилец № 2 = Бабка,
жилец № 3 = Внучка,
жилец № 4 = Жучка,
жилец № 5 = Кошка,
жилец № 6 = Мышка;
Глава № 2. ПОСАДИЛ ДЕД РЕПКУ. ВЫРОСЛА РЕПКА БОЛЬШАЯ-ПРЕБОЛЬШАЯ. СТАЛ ДЕД ЕЕ ТЯНУТЬ. ТЯНЕТ-ПОТЯНЕТ, А ВЫТЯНУТЬ НЕ МОЖЕТ.
Глава № 3. Сейчас номер жильца N = 1, а потом он будет меняться.
Глава № 4. Вспомните, чему равняется N, и к этой цифре прибавьте 1.
Глава № 5. ПОЗВАЛ ЖИЛЕЦ N—1 ЖИЛЬЦА N (ПРЕДЫДУЩИЙ ЖИЛЕЦ СЛЕДУЮЩЕГО ЖИЛЬЦА).
Глава № 6. ТЯНУТ-ПОТЯНУТ.
Глава № 7. Если N = 6, то переходите к главе № 10, иначе читайте дальше.
Глава № 8. А ВЫТЯНУТЬ НЕ МОГУТ.
Глава № 9. Возвращайтесь к главе №4 и читайте следующие за ней главы.
Глава № 10. ВЫТЯНУЛИ РЕПКУ!
```

Глава № 11. Конец сказки.

— Что же это за сказка? — воскликнул Сережа в недоумении. — Чепуха какая-то! Ой, извини, Чип, я хотел сказать, что сказка очень интересная, но только непонятная.

— Что же тут непонятного? — проворчал Чип. — Все понятно. Делай, что написано маленькими буквами, и читай, что написано большими, и получится сказка. Вот написано N=1, значит, запомни, что сейчас N равно 1. Написано — прибавь 1 к N, значит, посмотри, чему сейчас N равно, и прибавь 1. Было 1 — станет 2, потом, когда второй раз вернешься к главе № 4, будет 2, а станет 3. А вместо слов «жилец № такой-то» читай его имя по списку жильцов. Значит, на этот раз уже не дед бабку, а бабка внучку позовет. Теперь понял?

— Ну-ка, ну-ка, — до Сережи наконец все дошло. — Ох, как здорово! Значит, пока до жильца № 6, то есть до мышки, очередь не дойдет, я все буду возвращаться и возвращаться, и все новые жильцы будут репку тянуть. А на мышке все кончится, так заранее и рассчитано. Послушай, Чип, а ведь так можно хоть сто жильцов взять, и сказка длиннее не станет, только вместо 6 надо будет всюду сто писать.

— Вот сейчас, — провозгласил Чип торжественно, — ты понял самое главное в программировании: цикл. Все идет по кругу, только жильцы прибавляются. Так зачем много раз одно и

то же писать? Для этого и придуман цикл. Здесь цикл на мышке кончается, а бывает и бесконечный цикл, только его все компьютеры как огня боятся. Вот, например, сказка про попа и его собаку. Кстати, попробуй-ка напиши ее в виде программы.

— Сейчас попробую, — ответил Сережа. Он сел за стол и через пару минут написал:

ПОП И СОБАКА

```
Глава № 1. У ПОПА БЫЛА СОБАКА, ОН ЕЕ ЛЮБИЛ.
Глава № 2. ОНА СЪЕЛА КУСОК МЯСА.
Глава № 3. ОН ЕЕ УБИЛ.  И В ЗЕМЛЮ ЗАКОПАЛ.
Глава № 4. И НАДПИСЬ НАПИСАЛ:
Глава № 5. Если не надоело, возвращайтесь к главе № 1, иначе отдыхайте.
Глава № 6. Конец.
```

— Ого, — улыбнулся Чип, — да ты, я вижу, прирожденный программист. Пожалел компьютер и вставил условие «если не надоело» в пятой главе. Сделал лазейку из бесконечного цикла. Значит, если надоело, то можно отдохнуть.

— Знаешь, Чип, только мне эта сказка совсем не нравится. Разве можно собаку мучить? Ну, наказал бы, а то сразу убивать. Давай ее спасем!

— Давай, — охотно согласился Чип. — Пусть тот, кто читает, сам выбирает для собаки наказание. Вот заменим главу № 3 на такую:

```
«Глава № 3. ОН ЕЕ НАКАЗАЛ».
```

Теперь, перед тем, как читать сказку, надо выбрать наказание. Ну, скажем:

```
наказал = отругал и не дал мороженого,
```
или
```
наказал = отшлепал и запер на час в чулане.
```

Кстати, такая программа, в которой можно что-то определять по желанию перед тем, как пользоваться, называется подпрограммой. А то, что надо определять, пишется в скобках после названия. Значит, мы сейчас написали сказку-подпрограмму

```
ПОП И СОБАКА (наказал)
```

А когда будешь ее читать, можешь вставить, что хочешь, ну

хоть

```
ПОП И СОБАКА (поцеловал)
```

Видишь, здесь вставлено то, что надо читать вместо слова «наказал».

Ладно, спокойной ночи, мне пора. Завтра еще поиграем, хорошо?

ОТ РЕДАКЦИИ:

Дорогие ребята! Чип и вам дает задание: составить программу для сказки «Теремок». Самая лучшая и правильная программа будет напечатана в журнале. Ждем ваших писем.

Царевич выбирает невесту

На следующий день, когда Чип выпрыгнул из своей коробки, Сережа сразу его попросил:

— Давай сегодня еще в сказки-программы поиграем, а? Мне очень понравилось.

— Понравилось читать или понравилось писать?

— И то, и другое. Давай по очереди: сначала один, потом другой. Чур, ты первый пишешь!

— Ну ладно, только я ваших сказок совсем не знаю, я свою сочиню.

Чип думал минут двадцать, писал, зачеркивал, писал снова и даже что-то бормотал, забыв про все правила приличия для компьютеров. Наконец сказка-программа была готова. Вот что получилось.[2]

СКАЗКА-ПРОГРАММА «ЦАРЕВИЧ И КУХАРКА»

Жили-были царь с царицей, и был у них сын — красивый да разумный, одна беда — робкий слишком: никак невесту не мог себе найти. И вот позвали его царь с царицей в тронную залу и говорят: «Хватит тебе, Иванушка, холостым гулять, пора и жениться. Найди себе жену красивую, умную да работящую».

Пошел царевич прочь, закручинился: как самую лучшую отыскать? Найдешь самую умную, а вдруг урод? Выберешь самую красивую — лентяйка окажется. А самую работящую отыщешь — чего доброго, и дурочка, и уродина.

Идет-бредет он по дворцу и заходит в кухню. А там Ксюша-кухарка царскую посуду моет, песенку напевает. «Что, — говорит, — Иванушка, невесел, аль беда какая?» Говорит, как поет, а сама тарелки так и трет, зубки белые так и сверкают, глазки синие так и светятся.

«Эх, Ксюша, беда-то беда, да не бабьего ума это дело. А впрочем, послушай, может, что и посоветуешь».

Выслушала Ксюша царевича, подумала немного и говорит (а сама уже тарелки перемыла, за чашки принялась, руки белые так и мелькают): «Можно беде твоей помочь, Иванушка. Вот тебе программа выбора царской невесты.

```
Шаг №1. Отобрать по всему царству 512 самых красивых девушек;
шаг №2. из них отобрать половину самых умных;
шаг №3. из них отобрать половину самых работящих;
шаг №4. из них отобрать половину самых красивых;
шаг №5. если невест две или больше, вернуться к
```

шагу № 2;

шаг №6. если осталась одна невеста, сыграть свадьбу;

шаг №7. если осталось полневесты, значит, их было не 512, а, может, в суматохе кого-то потеряли. В этом случае вернуться к шагу №1;

шаг №8. конец программы».

«А за три дня-то успею?» — спрашивает Иванушка, а сам на Ксюшу смотрит не насмотрится.

«Успеешь, а как же, — отвечает Ксюша, а сама, как маков цвет, разрумянилась, косой закрывается от царевича, глаз на него поднять не может. — Успеешь, — говорит, — сердешный, я все рассчитала. За первый день 64 невесты останется, за второй — 8, а к третьему вечеру найдешь ты свою суженую: самую умную из красавиц, самую работящую из умниц и самую красивую из работящих».

Берет Иван-царевич за белу руку Ксюшу-кухарку, в сахарны уста целует и говорит ласково: «Спасибо, Ксюшенька, хорошо придумала, только не стану я три дня по всему царству искать, когда я уже здесь свое счастье нашел. Никого мне, кроме тебя, не надо, на тебе и женюсь!»

На том и порешили. Правда, царства им не досталось — крепко осерчали царь с царицей, — зато уж угощение на свадьбу Ксюша сготовила — гостей за уши не оттащишь!

Конец сказки».

— Ну, как? — гордо спросил Чип. — Не хуже, чем Мери Поппинс? Во всяком случае, программа у меня правильная — хоть невесту, хоть карандаш можно выбрать по трем признакам. Я сначала хотел ее так написать, чтобы признаков было не три, а сколько угодно, как жильцов в сказке о репке. Но потом подумал, что овчинка выделки не стоит: это во всем царстве невест не хватит, чтобы по десяти признакам выбирать. Так что пусть уж будет три признака, чтобы не просчитаться.

— Спасибо, Чип, замечательная сказка, — поблагодарил Сережа, — завтра моя очередь, а теперь спокойной ночи, а то уже поздно.

ОТ РЕДАКЦИИ:

А сегодня Чип просит наших читателей составить программу для сказки «Красная Шапочка». Лучшая работа будет напечатана в журнале.

Как спасти колобка

В воскресенье Сережа даже не вышел во двор. Ведь была его очередь сочинять сказку-программу. Он выбрал простую сказку: «Колобок». Но сделать из нее программу оказалось не так-то просто. Получался цикл, как в сказке про репку: добавляются новые звери, которые колобка хотят съесть. Сережа для них придумал название — «едоки». Вот что у него получилось.

СКАЗКА-ПРОГРАММА «КОЛОБОК»

Список едоков:
едок № 1 — дед,
едок № 2 — бабка,
едок № 3 — заяц,
едок № 4 — волк,
едок № 5 — медведь,
едок № 6 — лиса.

Глава № 1. ПРОСИТ ДЕД БАБКУ: «ИСПЕКИ МНЕ, СТАРАЯ, КОЛОБОК».

— А ГДЕ МУКИ ВЗЯТЬ-ТО?

— А ПО АМБАРАМ ПОСКРЕБИ, ГЛЯДИШЬ, И НАБЕРЕТСЯ.

Глава № 2. ИСПЕКСЯ КОЛОБОК НА СЛАВУ: КРУГЛЫЙ ДА РУМЯНЫЙ. ПОЛОЖИЛА ЕГО БАБКА НА ОКНО СТУДИТЬ.

Глава № 3. Начало цикла. Повторять для всех едоков подряд, начиная с зайца.

Глава № 4. ВЗЯЛ КОЛОБОК И УКАТИЛСЯ.

Глава № 5. КАТИТСЯ, КАТИТСЯ. А НАВСТРЕЧУ ЕМУ ОЧЕРЕДНОЙ ЕДОК: «КОЛОБОК, КОЛОБОК, Я ТЕБЯ СЪЕМ».

Глава № 6. «НЕ ЕШЬ МЕНЯ, ОЧЕРЕДНОЙ ЕДОК, Я ТЕБЕ ПЕСЕНКУ СПОЮ».

Глава № 7. «ПО АМБАРАМ Я СКРЕБЕН, В ПЕЧКЕ ПЕЧЕН, НА ОКОШКЕ СТУЖЕН, Я ОТ ВСЕХ ПРЕДЫДУЩИХ ЕДОКОВ УШЕЛ, А ОТ ТЕБЯ, ОЧЕРЕДНОЙ ЕДОК, И ПОДАВНО УЙДУ!»

Глава № 8. Конец цикла.

Вот тут Сережа и споткнулся. Получалось что-то не так. Ведь в сказке колобок поет очередному едоку не про всех предыдущих вместе, а про каждого из них в отдельности: зайцу поет про деда и бабку, волку — про деда, бабку и зайца, и так далее. Чем дальше катится, тем длиннее песенка, пока лиса не попадается.

Сережа мучился, мучился, а потом не выдержал и позвал на помощь Чипа.

— Ну что ж, я вижу, ты делаешь успехи, — похвалил его Чип,

— надо только кое-что исправить. Вот как нужно сделать, чтобы он спел не про всех едоков вместе, а про каждого по отдельности:

Глава № 7. «ПО АМБАРАМ Я СКРЕБЕН, В ПЕЧКЕ ПЕЧЕН, НА ОКОШКЕ СТУЖЕН...»
Глава № 8. Начало второго цикла. Повторять для всех предыдущих едоков, начиная с деда.
Глава № 9. «Я ОТ ПРЕДЫДУЩЕГО ЕДОКА УШЕЛ...»
Глава № 10. Конец второго цикла.
Глава №11. «...А ОТ ТЕБЯ, ОЧЕРЕДНОЙ ЕДОК, И ПОДАВНО УЙДУ!»
Глава № 12. Конец цикла.

— Чем сказка кончается, — спросил Чип, — лиса его съест, так что ли?

— А это необязательно — ответил Сережа, — мы сочиним новую сказку, где никто колобка не съедает, а все вместе поют его песенку.

— Ну что ж, тогда напишем так:

Глава № 13. Если вам жалко колобка, то спойте с ним его песенку.
Глава № 14. Если вам его не жалко, то возьмите книжку и прочитайте, как его лиса съела. Нам про это писать не хочется.
Глава № 15. Конец.

— Вот и все, — Чип снял шапочку и раскланялся. — А вообще-то ты сложную сказку выбрал, с двойным циклом. Каждому новому едоку колобок поет про каждого предыдущего. Вот обрати внимание на главу № 9. Она повторяется 2 раза для зайца, 3 раза для волка, 4 раза для медведя и 5 раз для лисы — итого 14 раз. Помнишь, я тебе говорил, что цикл придумали, чтобы много раз одно и то же не писать. А двойной цикл еще больше места экономит. Представь себе, что было бы, если бы колобок сто разных зверей встретил, пока ему лиса не попалась. Наша программа почти не увеличится — надо только список едоков расширить, а обычная сказка про колобка, знаешь, как увеличится! Мало того, что нужно будет описать встречу с каждым из сотни зверей, нужно будет, чтобы колобок каждому зверю пропел про каждого предыдущего. На сказку, небось, и целого журнала не хватит. А мы с тобой на одну страничку уместились.

— Так что же, цикл только для экономии бумаги придуман?

— спросил Сережа, которому немножко надоело хвастовство Чипа.

— Да что там бумага, — Чип махнул рукой, — не на бумаге же компьютерам программы пишут. У нас и магнитные ленты, и диски вроде грампластинок, и специальные кристаллики для памяти, и все равно памяти не хватает. Сейчас мы можем делать миллионы и даже миллиарды операций в секунду, и работаем по нескольку дней без остановок. Вот и представь, что было бы, если бы каждую операцию надо было отдельно описывать. А так написал цикл: «Сто миллиардов раз сложи 2 и 2», и пожалуйста! Компьютер работает, а ты отдыхаешь. Не пишешь ему сто миллиардов раз, чтобы он 2 и 2 сложил.

Ну, ладно, хватит на сегодня. В воскресенье и погулять надо.

ОТ РЕДАКЦИИ:

А сегодня вам Чип дает задание, ребята, написать программу для любимой считалочки. Лучшую программу мы напечатаем.

Случай в квартире 130

— Во что мы сегодня будем играть? — нетерпеливо спросил Сережа, как только Чип выпрыгнул из коробки.

— Во что, во что... Будто я массовик-затейник — проворчал Чип, но по его хитрым глазкам Сережа понял, что Чип приготовил ему сюрприз и ломается только для вида.

— Ну, пожалуйста, Чип, миленький, я же знаю, что

ты все можешь придумать!

— Ну уж все... Ты в «Джеков дом» умеешь играть?

— Какой такой «Джеков дом»? Я такой игры не знаю.

— Неужели не слышал стишок: «Вот дом, который построил Джек. Вот пшеница, которая в темном амбаре хранится, в доме, который построил Джек...» И так далее. Все новые строчки прибавляются, а старые повторяются.

— А... слышал, но какая же тут игра?

— А вот какая: давай сочинять похожий стишок. Я говорю первый куплет, ты в ответ второй, я третий, ты четвертый и так далее, пока кто-то не сдастся. Ну, например, у вас дома кто строит?

— Не знаю, кажется, какой-то ЖЭК. [3]

— Ну, тогда начнем так: «Вот дом, который построил ЖЭК». Твоя очередь, Сережа.

— «Вот квартира сто тридцать,

В которой неладное что-то творится,

В доме, который построил ЖЭК», — сказал Сережа, немного подумав. — Твой ход, Чип.

— Э... э, назови-ка мне какое-нибудь имя девочки.

— Аня, — назвал Сережа имя своей лучшей подруги.

— Ну тогда: «Вот девочка Аня, которая спит у себя на диване,

В квартире и т.д.». Твоя очередь, Сережа. Так что же там неладное творится? Придумывай.

Сережа задумался.

— «Вот комната ванная, в которой море шумит разливанное,

Из крана, забытого девочкой Аней, и т.д.».

— Теперь назови мужское имя, — попросил Чип.

— Мужское? Пожалуйста: Никита.

— «Вот слесарь Никита по лестнице мокрой шагает сердито:

Спешит он в ту комнату ванную и т.д.», — не задумываясь, выпалил Чип.

— Ладно, сдаюсь! — со смехом сказал Сережа.

— А теперь, — сказал Чип вкрадчиво, — как ты, наверно, догадался, мы сделаем из этого стишка программу.

— Подумаешь, легкота! Мы такое уже делали.

— Вот как? Ну, попробуй напиши программу хотя бы для первых трех куплетов.

Сережа взялся за дело и довольно скоро понял, что тут что-то не то. В старых сказках-программах повторялись едоки или жильцы, то есть отдельные слова. А тут повторялись целые куплеты, да еще при этом менялись падежи слов внутри куплетов.

— Что, не тянется репка? — посочувствовал Чип. — А помнишь, мы с тобой про попа и собаку подпрограмму сочиняли? [4] Там ведь можно было выбрать любое слово: «поцеловал», «наказал» — и вставить его внутрь подпрограммы. Вот так и здесь надо. Например:

```
Подпрограмма «ДОМ».
ДОМ, КОТОРЫЙ ПОСТРОИЛ ЖЭК.
Возврат.
```

Выделенное слово ДОМ будет склоняться так, как тебе нужно: дом, доме, домом и так далее.

```
Подпрограмма «КВАРТИРА».
КВАРТИРА СТО ТРИДЦАТЬ, В КОТОРОЙ НЕЛАДНОЕ ЧТО-ТО ТВОРИТСЯ, В «ДОМЕ»...
Возврат.
```

— Видишь, — продолжал Чип, — тут в кавычках написано слово «ДОМЕ». Это значит, что вместо него надо вставить подпрограмму «ДОМ», то есть написать: «в доме, который построил ЖЭК». Ну, а дальше так же.

```
Подпрограмма «ДЕВОЧКА АНЯ».
ДЕВОЧКА АНЯ, КОТОРАЯ СПИТ У СЕБЯ НА ДИВАНЕ, В «КВАРТИРЕ»...
Возврат.
```

На этот раз вызывается подпрограмма «КВАРТИРА», то есть: «квартире 130, в которой неладное что-то творится в «доме». Обрати внимание, что подпрограмма «КВАРТИРА», в свою очередь, вызывает подпрограмму «ДОМ». Понятно?

— Вроде да, — сказал Сережа неуверенно, только я не понимаю, зачем все время пишется слово «возврат» в конце подпрограммы.

— А как же, это значит, что надо вернуться к тому месту, откуда вызывалась подпрограмма, и продолжать дальше.

Например, после того, как закончится подпрограмма «ДОМ», надо продолжать подпрограмму «КВАРТИРА», а когда она закончится, надо продолжать подпрограмму «ДЕВОЧКА АНЯ». Ну как, сможешь дальше сам?

— Попробую. — ответил Сережа и скоро написал:

```
Подпрограмма «КОМНАТА ВАННАЯ».
КОМНАТА ВАННАЯ, В КОТОРОЙ МОРЕ ШУМИТ РАЗЛИВАННОЕ,
ИЗ КРАНА, ЗАБЫТОГО «ДЕВОЧКОЙ АНЕЙ»...
Возврат.
```

— Правильно, — похвалил Чип. — А вот, наконец, последняя подпрограмма.

```
Подпрограмма «СЛЕСАРЬ НИКИТА».
СЛЕСАРЬ НИКИТА ПО ЛЕСТНИЦЕ МОКРОЙ ШАГАЕТ СЕРДИТО:
СПЕШИТ ОН В ТУ «КОМНАТУ ВАННУЮ»...
Возврат.
```

— А все стихотворение можно записать так, — сказал Чип:

```
Программа «СЛУЧАЙ В КВАРТИРЕ 130».
Вот «ДОМ»; вот «КВАРТИРА»; вот «ДЕВОЧКА АНЯ»;
вот «КОМНАТА ВАННАЯ»; вот «СЛЕСАРЬ НИКИТА».
Конец.
```

Можешь проверить, расписав каждую подпрограмму.

Новая игра

Многие мальчики и девочки спрашивают в письмах Чипа: не знает ли он игры, в которую можно поиграть на самом простом калькуляторе?

Знает Чип такие игры. Вот одна из них:

«ЧИСЛОВЫЕ ПРЫГАЛКИ»

На калькуляторе набирают любое число меньше 100. Двое играющих «ходят» по очереди: если число четное, то его в один ход делят пополам. Если нечетное, то другой игрок, тоже за один ход, сначала число умножает на 3, а дальше по своему усмотрению или отнимает, или прибавляет единицу. Выигрывает тот, кто в ответе получает единицу.

Вот как протекала одна партия между Сережей и его подругой Аней.

Аня набрала на калькуляторе число 5.

Сережа: 5x3-1 = 14.

Аня: 14:2 = 7.

Сережа: 7x3-1 = 20

Аня: 20:2 = 10.

Сережа: 10:2 = 5.

Аня: 5x3 + 1 = 16.

Сережа: 16:2 = 8.

Аня: 8:2 = 4.

Сережа: 4:2 = 2.

Аня: 2:2 = 1.

Сережа проиграл, ему осталась единица.

Счет в этой игре можно вести и по-другому: суммировать числа, полученные в результате хода каждого игрока. Выиграет тот, кто наберет больше очков к концу игры, когда в ответе получится единица.

Аня: 5 + 7 + 10 + 16 + ... = 43.

Сережа: 14 + 20 + 5 + 8 + ... = 49.

А так счет 43:49 в пользу Сережи.

ОТ РЕДАКЦИИ:

Ребята, Чип дает вам задание: подумать, какие ходы могут привести вас к победе. Подсказываем, у каждого варианта игры — свои секреты. Кто догадается, напишите нам.

Барон Мюнхгаузен и урок физкультуры

— Ты над чем так смеешься? — спросил Чип. Сережа зачитывался «Бароном Мюнхгаузеном», а Чип прогуливался по столу, поглядывая по сторонам. Работы для него не было, и он томился от безделья.

— Да вот, представляешь, барон Мюнхгаузен сам себя из лужи вытянул! За волосы!

— И что же тут смешного? По-моему, ничего особенного.

— Уж будто твои программы могут сами себя за волосы тащить!

— А что! — азартно крикнул Чип. — Вот спорим, что я напишу программу, которая сама себя из лужи вытаскивает? Спорим?

— Ну, спорим, — усмехнулся Сережа.

— Это будет программа

```
«МЮНХГАУЗЕН».
«ЕСЛИ МЮНХГАУЗЕН В ЛУЖЕ, ТО ОН ДОЛЖЕН ТЯНУТЬ СЕБЯ ЗА ВОЛОСЫ».
```

А «ТЯНУТЬ СЕБЯ ЗА ВОЛОСЫ» — это подпрограмма.

```
Подпрограмма «ТЯНУТЬ СЕБЯ ЗА ВОЛОСЫ»:
«НАГРЕТЬ СВОИМИ ДВИЖЕНИЯМИ ЛУЖУ НА ОДНУ ТЫСЯЧНУЮ ГРАДУСА. ЕСЛИ ЛУЖА НЕ ВЫСОХЛА, СНОВА ТЯНУТЬ СЕБЯ ЗА ВОЛОСЫ».
Конец подпрограммы.
```

— Ну, а как же эта подпрограмма поможет Мюнхгаузену вылезти из лужи? — спросил Сережа с подозрением.

— А вот как: Мюнхгаузен первый раз потянет себя за волосы и нагреет своими движениями лужу на одну тысячную часть градуса. Потом, если лужа не высохла (а она, конечно, и не подумала высохнуть, хоть чуть-чуть и нагрелась), он снова потянет и нагреет лужу еще на одну тысячную часть градуса и так далее. Если у Мюнхгаузена скорость работы, как у рядового компьютера, то не пройдет и секунды, как вода в луже поднимается до 100°С и закипит. Ну, а из кипятка-то, я думаю, он и сам не заметит, как выпрыгнет!

— Здорово, — восхищенно протянул Сережа. — Слушай, а я

как-то и не понимал, что компьютеры так быстро работают. В одну секунду: раз — буль-буль — и прыг! Готово дело!

— А ты заметил другое, что моя подпрограмма вызывает сама себя, как и Мюнхгаузен сам себя за волосы тащит? И не просто себя вызывает. Если бы она только себя вызывала и больше ничего не делала, то получился бы бесконечный цикл, как в стишке про попа и собаку. Но она при каждом вызове чуть нагревает лужу, так что рано или поздно лужа испарится и цикл закончится.

Такие подпрограммы называются рекурсивными, их всегда трудно понять. Кажется, что они не смогут работать, как Мюнхгаузен не сможет вытянуть себя из лужи. Для того, чтобы рекурсивная подпрограмма работала, надо, чтобы при каждом вызове что-то изменялось так, чтобы работа могла кончиться.

— А можешь еще рекурсивную программу написать? Мне очень понравилось.

— Ну что ж, например, ваш класс на физкультуре неправильно выстроился по росту: впереди самый маленький, в затылок ему смотрит мальчик повыше, а сзади самый большой. Вот такая рекурсивная программа перестраивает их в обратном порядке.

```
Программа «ПЕРЕСТРОЙ» (колонну).
Если в колонне один человек, то возврат.
ПЕРВЫЙ ДЕЛАЕТ ШАГ ВБОК.
«ПЕРЕСТРОЙ» (остаток колонны).
ПЕРВЫЙ ИДЕТ НАЗАД.
Возврат.
```

— Постой, постой, дай я соображу, как она будет работать. Если в колонне всего один человек, то его нечего перестраивать. Правильно, тут и написано «возврат». А куда возврат? Ладно, пока это неважно, а там посмотрим.

Теперь пусть в колонне два человека. Идем по программе. Первая строчка не для нас, поскольку в колонне два человека.

Смотрим следующую строчку. Первый делает шаг вбок — это мы выполнили вторую строчку. Согласно третьей строчке, мы должны перестроить остаток колонны, то есть последнего, второго. Это мы уже умеем — он остается на месте, и происходит какой-то возврат. Чип, а что это за возврат?

— Ну, сам подумай, что ты должен сейчас делать? Вспомни, что ты делал до того, как прочел таинственное слово «возврат», и... **возвращайся** к этому делу.

— Как что делал? Я перестраивал колонну из двух человек. У меня первый сбоку стоит и ждет.

— Вот и возвращайся к тому, что ты дальше должен с ним делать. В программе на четвертой строчке сказано: «Первый идет назад». Это как раз к нему и относится.

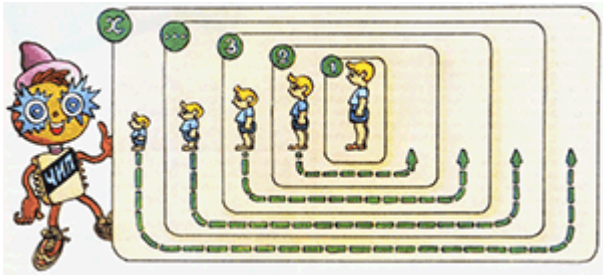
— Так, значит, первый идет назад и оказывается позади второго. Опять возврат! Колонна из двух человек перестроена. Все-таки я с этим возвратом чего-то не понял. Интересно, как он будет работать с тремя людьми? Первая строчка нашей программы для колонны из трех человек не годится. На второй строчке первый человек делает шаг вбок, а на третьей... Ага, на третьей строчке мы перестраиваем колонну из двух, это мы уже умеем. Перестроили. На четвертой строчке ставим первого сзади них, то есть на самое последнее место, как и надо.

А зачем же нужен возврат? Ну, теперь ясно — от перестройки одного человека мы вернулись к перестройке двоих, от перестройки двоих к троим и так далее. Значит, возвращаемся на один шаг назад, к самому последнему из брошенных дел. Чип, а знаешь, что мне еще непонятно: как это программа будет работать, если взять много людей, ну, скажем, сто? На третьей строчке написано: «Перестрой остаток колонны», а там 99 человек. Мы же еще не знаем, как их перестраивать?

— А и не надо заранее знать! Начинай применять к ним ту же самую программу, и все получится. Они по очереди будут делать шаг вбок, пока не дойдет черед до последнего, а потом предпоследний встанет позади последнего, предпоследний за ним и так далее, пока первый не встанет сзади всех. Программа будет применяться сто раз, и сто раз будет происходить возврат к предыдущей перестройке.

— Как сложно, правда, Чип? Три строчки написано, а сколько беготни.

— А ты попробуй к следующему разу сам сочинить рекурсивную программу. Договорились?



ОТ РЕДАКЦИИ.

Ребята, может быть, и вы попробуете сочинить рекурсивную программу для сказки или считалки? А Чип проверит, что у вас получится. Ждем ваших писем.

Конкурс поросят

К следующему разу Сережа ничего путного не смог придумать. В голову лезла какая-то чепуха про двенадцать поросят.

«12 поросят на палубе сидят.
Они песенки поют, им уроки задают.
Один из них устал и с палубы удрал,
И вот результат — 11 поросят.
11 поросят на палубе сидят...».
И так далее...

В конце концов из этого тоже можно сделать программу. Сережа взял листочек бумаги и написал:

```
Программа. «ПЕСЕНКА ПРО N ПОРОСЯТ».
```

Но тут вспомнил, как Чип объяснял ему: «Если внутри программы что-то меняется, то это уже не программа, а подпрограмма».

«У нас же число поросят будет меняться», — подумал Сережа, зачеркнул первую строчку и написал вот что:

```
Подпрограмма.
«ПЕСЕНКА ПРО N ПОРОСЯТ»
Если N=0, то конец.
N ПОРОСЯТ НА ПАЛУБЕ СИДЯТ.
ОНИ ПЕСЕНКИ ПОЮТ, ИМ УРОКИ ЗАДАЮТ
ОДИН ИЗ НИХ УСТАЛ И С ПАЛУБЫ УДРАЛ.
И ВОТ РЕЗУЛЬТАТ: N-1 ПОРОСЯТ
«ПЕСЕНКА ПРО N-1 ПОРОСЯТ»
Конец подпрограммы.
```

— Ну что, не так уж плохо, — сказал Чип, прочтя про поросят. — Не так плохо для начала: ты догадался, что нужно написать не программу, а подпрограмму и остановиться, когда все поросята удерут с палубы.

— Ну, а что у тебя? — поинтересовался Сережа.
— А у меня конкурс для твоих поросят.
— Конкурс? Какой конкурс?

— Ну, скажем, по пению. Они песенки поют? Вот мы и проверим, кто лучше поет.

```
Подпрограмма «Лауреат... (среди поросят)».
Если поросенок один, то объявить его лауреатом.
Если их больше, то отвести в сторону первого поросенка и найти лауреата среди остальных поросят.
Сравнить пение этого лауреата и первого поросенка.
Объявить победителя лауреатом.
Конец.
```

— Ага, — Сережа задумался, — это похоже на перестройку колонны, которую мы делали в прошлый раз. Сейчас я вспомню. Значит, сначала мы поросят будем поочередно отводить в сторону, потом начнем с конца: объявим последнего лауреатом, сравним его с предпоследним, и так до самого первого. Вспомнил, вспомнил! Знаешь, Чип, а это несправедливо: последнему дали побыть лауреатом, даже не проверяя, как он поет. А бедненький первый, даже если он по пению на втором месте, ни разу лауреатом не был.

— Ничего не поделаешь, — возразил Чип. — Мы же хотели найти только самого лучшего. Если ты хочешь всем дать по заслугам, надо теперь нашего лауреата в сторону отвести и найти лауреата среди остальных. Так ты найдешь второго призера, отведешь его в сторону и найдешь третьего, ну, и так далее. Для этого нужна такая подпрограмма:

```
Подпрограмма «Конкурс поросят на приз N»
Найти «Лауреата» (среди поросят).
Дать ему приз N.
Если поросята кончились, то конец.
Провести «Конкурс (оставшихся поросят) на приз N + 1».
Конец.
```

Видишь, эта подпрограмма вызывает старую подпрограмму «Лауреат (среди поросят)», а потом еще вызывает сама себя. Вот так она и будет находить оставшихся лауреатов и давать им положенные призы.

— Чип, слушай-ка, — спохватился Сережа, — а что это мы с тобой стараемся ради этих глупых поросят? Ну, какая разница, кто из них лучше поет?

— Вот как? А ты знаешь, что эта задача — как быстрее

провести конкурс — нужна не только для поросят. Она очень нужна для людей, и ею занимаются лучшие программисты и математики во всем мире. Ведь человеку в любой деятельности часто приходится с помощью компьютеров сортировать те или иные предметы по какому-то признаку — величине, яркости и так далее. И чем больше предметов, тем больше времени занимает сортировка, тем она труднее и дороже. Если научиться ее делать в два раза быстрее, то во всем мире будет сэкономлено столько труда, что можно будет накормить сотни голодных детей где-нибудь в Африке. А ты говоришь, поросята. Эх ты!

— Извини, Чип, я же не знал. А скажи, наша программа — она быстро работает?

— Да нет, есть и более быстрые, только их так просто не напишешь. Это целая наука — сортировка. Ну ладно, мы заговорились, а уже пора спать.

ОТ РЕДАКЦИИ:

Ребята, в этом и предыдущем номерах журнала Чип объяснял вам очень сложные вещи, которые и взрослые не сразу понимают. Но вы не смущайтесь, сыграйте в перестройку колонны, в конкурс поросят. Ведущий делает все по программе, как компьютер, а остальные его слушаются. Для начала в игре пусть участвуют два-три человека, а потом можно проверить программу и для целого класса. Если выполнять программу точно и не сбиваться, все у вас получится. Только тянуть себя за волосы из лужи не надо.

Сегодня мы называем имена тех наших читателей, которые первыми прислали правильные программы к сказке «Теремок». Это Покрас Марианна, г.Москва: Варонанский Сергей, г.Москва: Денисов Витя, п.Протвино, Московская обл.; Трахтман Жанна, г. Свердловск; Венкина Катя, г. Москва; Мельникова Оля, г. Киев; Моор Коля. г. Березняки; Молотков Слава, г. Горький; Евстронова Надя, г. Свердловск; Ухова Галя, г. Троицк, Челябинская обл.; Сахбутдино-ва Лейла, г.Казань. Молодцы ребята!

Вот программа, составленная пятиклассником Сережей Варонанским:

```
Сказка-программа «Теремок».
Глава № 1. Жили-были:
жилец № 1 — Мышка-норушка,
жилец № 2 — Лягушка-квакушка,
жилец № 3 — Петушок—золотой гребешок,
жилец № 4 — Зайчик-побегайчик,
жилец № 5 — Лисичка-сестричка,
жилец № 6 — Волк—зубами щелк,
жилец № 7 — Медведь — мастер реветь.
Глава № 2. СТОЯЛ В ПОЛЕ ТЕРЕМОК.
Глава № 3. БЕЖАЛА МИМО МЫШКА-НОРУШКА. ЗАГЛЯНУЛА И ОСТАЛАСЬ ЖИТЬ.
Глава № 4. Сейчас номер жильца N=1, а потом он будет меняться.
Глава № 5. Вспомните, чему равняется N. и к этой цифре прибавьте 1.
Глава № 6. ШЕЛ МИМО ЖИЛЕЦ N. ЕГО ПОЗВАЛИ ПРЕДЫДУЩИЕ ЖИЛЬЦЫ. И СТАЛИ ОНИ ВМЕСТЕ ЖИТЬ.
Глава № 7. Если N равно 7, то читайте дальше, иначе возвращайтесь к главе №5.
Глава № 8. ЖИЛЕЦ № 7 РАЗДАВИЛ ТЕРЕМОК
Глава № 9. Конец сказке.
```

Электронная яблоня

Октябрьский воскресный день начинался безрадостно. Сережа и Чип уныло глядели в окно — небо хмурилось, того и гляди дождь пойдет. Делать было нечего, все игры по десять раз переиграли, скучно!

— Чип, а ты не можешь наколдовать, чтобы погода исправилась? — лениво протянул Сережа.

— Ну, нет, над вашей погодой я не властен, вот с электронами я что хочешь наколдую.

— Что хочу? Прекрасно! Наколдуй в телевизоре, чтобы получилась специально для нас с тобой интересная передача. Ведь в телевизоре электроны? Скажешь, нет?

Чип немного растерялся. Видно было, он не знает, что возразить, а признать себя побежденным ему не позволяет гордость. Но колебался он недолго, сверкнул глазами, нахлобучил свою шапочку, ринулся к работающему телевизору и, прежде чем Сережа успел его остановить, исчез за задней стенкой.

По экрану побежали полосы, раздалось гудение, и вдруг вместо диктора появилось улыбающееся личико Чипа.

— Готово! — весело крикнул Чип с экрана. — Теперь-то мы с тобой поиграем на славу! [5] Вот этот маленький человечек будет за тебя. — Чип показал на смешного, карикатурного мальчика рядом с собой. Тот помахал Сереже рукой и улыбнулся до ушей. — Управлять ты им будешь с помощью ручек настройки телевизора. Поверти их и увидишь, что человечек будет двигаться, куда ты захочешь. А уж говори за него сам.

Сережа повертел ручки и скоро понял, как управлять мальчиком и как двигать его руками.

— Хватит без дела слоняться. — Чип взял мальчика за руку и повел за собой. — Пошли искать приключения! Кстати, — Чип обернулся и подмигнул Сереже, — посмотри, какая в моем электронном царстве прекрасная погода!

Действительно, на экране был замечательный день. Светило солнце, пели птички, Чип и мальчик шли по яблоневому саду.

Недолго думая, Сережа направил мальчика к яблоне и заставил на ходу сорвать самое румяное яблоко. И тут, как это часто бывает, когда нечаянно сорвешь яблоко, появился сторож.

— Я только попробовать хотел, — оправдывался Сережа за мальчика, который стоял, повесив голову перед грозным сторожем.

— Ах вот оно что, — пробасил сторож, — ты сорвал самое

спелое яблоко, — значит, ты разбираешься в яблонях? Сейчас мы это проверим. — Он вынул из кармана черный шелковый платок и завязал мальчику глаза. — Вот тебе корзинка, полезай-ка на яблоню и собери все яблоки. Да смотри не подглядывай! А пропустишь хоть одно, полезай на следующее дерево и работай, пока не научишься.

Чип обернулся к Сереже, открыл рот, чтобы что-то сказать, но сторож так погрозил пальцем, что Чип прикусил язык и только умоляюще смотрел: не подкачай, мол.

Сережа подвел мальчика к дереву, и тот полез вверх, держа в одной руке корзинку. Изображение выросло во весь экран, так что было видно только место на яблоне, по которому полз мальчик. Вот он дополз до развилки.

«Так, — подумал Сережа, — полезем налево, там, помнится, было больше яблок. Только бы потом не забыть и про правую ветку».

Только на пятой развилке Сережа понял, что так дело не пойдет. Невозможно упомнить, на какую ветку мальчик уже забирался, а на какую — нет. Вконец замучившись, Сережа спустил мальчика с дерева. Сторож злорадно захохотал. Теперь, когда изображение уменьшилось, стало видно, что Сережа с мальчиком пропустили одну ветку, полную яблок.

Сережа задумался: «Прежде чем посылать мальчика на вторую яблоню, надо составить план. Вот полезет он на яблоню и вдруг — развилка. Тут надо действовать по порядку, чтобы ничего не спутать. Залез на одну ветку — обери ее до конца и уж только потом переходи к следующей. А на той ветке, где яблок уже не осталось, ниточку повяжи. Тогда хоть десять веток у тебя на развилке — все равно не перепутаешь. Спустился, например, с седьмой ветки, повязал ее у ствола ниточкой и оглядись: сколько еще не повязанных веток на этой развилке осталось. Ага, вот эти три, возьми любую и лезь на нее да яблоки собирай.

Итак, чтобы собрать яблоки со всего дерева, нужно добраться до первой развилки и поочередно обобрать каждую ветку. А как обобрать каждую ветку? Да точно так же: доползти до ее первой развилки и обобрать каждую из этих веточек. И так далее, пока ветки не кончатся. Смешно получается: я знаю, что мне делать на каждой развилке, но не могу заранее сказать, какое яблоко раньше сорву. А смогу я это сказать, только когда сверну к этому яблоку на последней развилке».

Сережа попросил у сторожа моток ниток и уверенно направил мальчика к дереву. На этот раз он не пропустил ни одного яблока. Даже строгий сторож улыбнулся, погладил мальчика по голове и подарил ему собранную корзинку яблок.

Выбравшись из телевизора, Чип гордо заметил: «Все-таки не зря я тебя учил рекурсивным сказкам и стишкам».

— А при чем тут они? — удивился Сережа. — Я просто составил план: добраться до первой развилки и по такому же плану просмотреть каждую ветку.

— Да это и есть рекурсивная программа, вернее, подпрограмма, потому что ее можно вставлять внутрь любой программы или подпрограммы. Вот как воспел ее один поэт, пожелавший остаться неизвестным.

Чип смущенно покашлял, принял торжественную позу и продекламировал:

```
Рекурсивный стих-подпрограмма
ОБИРАЕШЬ (ветку)
По стволу ты полезай, видишь яблоко — хватай.
На развилку вылез вдруг — ОБИРАЕШЬ (каждый сук).
Яблоки собрал — ВОЗВРАТ: лезешь по стволу назад.
```

Здесь все дело во второй строчке. Если ветка разветвляется, то и строчку тоже надо разветвить: заменить слова ОБИРАЕШЬ (каждый сук) на весь стишок для каждого сучка. Сколько сучков, столько и стишков. Если и те разветвятся, то и стишки размножатся. Вот смотри, как это нарисовал художник.

— Знаешь, — продолжал Чип, — после цикла «дерево» — самое важное в программировании. Конечно, не яблоня и не дуб, а «дерево», как схема выбора. Подпрограмма, которую мы составили, так и называется во всех учебниках: «обход дерева». Все программы, которые управляют сложными процессами, например, ведут воздушный бой или играют в шахматы, перебирают варианты, как и при сборе яблок с дерева. А компьютеры будущего, ученые называют их компьютеры пятого поколения, смогут сами перебирать варианты. Скажешь им:

«Собери все яблоки», — и они будут автоматически обходить все ветки, не пропуская ни одной.

Ну ладно, — закончил свои объяснения Чип, — небо-то прояснилось, надо тебе и погулять.

ОТ РЕДАКЦИИ:

Ребята, сегодня Чип вам дает задание составить рекурсивную подпрограмму для сбора плодов с дерева манго.

Представьте тропический лес, деревья, опутанные лианами, усыпанные гнездами попугаев. Лианы спускаются на землю, свисают над водой, а где-то оказывается, что это не лианы даже, а змеи...

Как собрать с дерева все плоды и не набрать в корзинку птичьих гнезд?

Один шаг Чип вам подскажет:

«Если сполз по ветке на землю, снова иди к дереву».

Лучшие подпрограммы будут напечатаны. На конверте напишите название задания: «ПРИКЛЮЧЕНИЕ В ДЖУНГЛЯХ».

Как уговорить маму купить жирафа

В этот вечер Сережа был занят: родители ушли в гости, а ему поручили домашнюю работу. Надо было убрать в кухне, вымыть посуду и почистить картошку. Чтобы было не так скучно, он вызвал Чипа, и теперь тот иронически комментировал Сережины действия:

— Ну как ты моешь пол?! То тут, то там машешь тряпкой без толку, а вода остывает. Вот этот угол ты уже третий раз трешь, а про тот забыл.

— Может быть, и на этот случай ты программу составишь? — отозвался Сережа, пыхтя и возя мокрой тряпкой по полу.

— Пожалуйста, вот тебе рекурсивная подпрограмма:

```
ВЫМОЙ (пол).
Если пол вымыт, то возврат.
Протри тряпкой полоску от левого дальнего до правого дальнего от двери угла.
ВЫМОЙ (оставшуюся часть пола).
```

По этой программе ты будешь мыть пол, как ты читаешь страницу: строчка за строчкой, начиная с левого дальнего угла и кончая правым ближним к двери. Через дверь потом выйдешь, чтоб по мокрому не ходить.

— Ладно, — усмехнулся Сережа, — когда у меня будет слуга-робот, я ему напишу эту программу. И еще добавлю строчку, чтобы он столы и стулья отодвигал, а то ты про это забыл. Слушай, Чип, а что, программу можно для всего на свете составить?

— Составить-то можно, а вот как она будет работать, это еще надо посмотреть. Ну вот что бы ты хотел?

— Скажем, я хотел бы, чтобы мама мне купила жирафа!

— А что. — задумался Чип. — под окном привяжешь, у вас второй этаж, можно прямо через окно кормить. Словом, никаких хлопот и море удовольствия. Осталось только маму уговорить. Да, тут нужна очень хорошая программа. Это будет программа для тебя: как ты должен себя вести, чтобы мама купила жирафа. У тебя есть идеи, что может на нее подействовать?

— Ну, скажем, — протянул Сережа, все больше увлекаясь своей шуткой, — скажем, я дождусь, когда она меня будет за что-нибудь ругать, что я руки мыть перед едой забываю или что я поздно спать ложусь, а я тут возьму да и скажу: «А вот спорим, я за всю четверть больше ни разу так не сделаю?» Она скажет:

«Свежо предание, да верится с трудом». А я в ответ: «Ну что, спорим?» А она спросит: «А на что?» Я в ответ: «Да хоть на жирафа». А она возьмет и в шутку согласится!

— Ну, а где же вы его возьмете, этого жирафа?

— А я поступлю в кружок юннатов при зоопарке, и, когда маленький жирафенок будет болеть, я с ним ночи не буду спать, выхожу, а потом...

— Слушай, Сережа, а у меня другая идея: давай предложим всем ребятам эту задачу. Пусть наши читатели попробуют составить программу, как уговорить маму купить жирафа. У всех разные мамы, и каждый придумает свой способ. Все дело в том, чтобы записать его в виде программы последовательных действий, вроде той, которую мы начали составлять.

— И еще, — подхватил Сережа, — пусть попробуют написать программу для чистки картошки, вроде той, которую ты предложил для мытья пола.

— А последнее — пусть запрограммируют стишок, сказку или считалку по своему выбору, вроде тех, которые мы с тобой раньше сочиняли. Только пусть хорошенько проверят, правильно ли программа работает, прежде чем посылать ее нам. Советуем всем ребятам дождаться следующего номера журнала, где мы с тобой будем разбирать их письма со сказками «Теремок» и «Красная Шапочка», а уже потом присылать свои программы.

ОТ РЕДАКЦИИ:

*Итак, журнал «Пионер» вместе с Чипом и Сережей объявляет конкурс на лучшие программы, составленные нашими читателями самостоятельно, **без помощи взрослых**. В конкурсе три задачи:*

1. Как уговорить маму купить жирафа.

2. Программа чистки картошки.

3. Программа любимого стишка, сказки или считалки. ВНИМАНИЕ! Победители будут награждены.

Первый приз — КАЛЬКУЛЯТОР! Решения вы должны выслать ***до 1 января 1987 года****.*

Не забудьте указать свой точный адрес, возраст, имя и фамилию. На конверте поставьте пометку: «КОНКУРС ЧИПА».

Чип и Сережа читают ваши письма

Чип и Сережа разбирали письма, пришедшие в журнал «Пионер», целое воскресенье. Это было так интересно! Письма были разные, из всех республик, от старших и младших школьников. Оказывается, многих заинтересовали сказки-программы, и очень многие успешно справились с заданием составить программу по сказке «Теремок».

В № 9 мы напечатали имена ребят, которые первыми прислали правильные программы. Сегодня мы можем этот список продолжить. Интересные и правильные программы составили: Лена Алексеева, г. Мозырь; Таня Ковалева, г. Гомель; Оля Котенко, г. Киев; Д. Угай, Ленинградская область; Гузель Гильфанова, г. Лениногорск; Ира Ионова, Крымская область; Юля Лерман, г. Ленинград; Алексей Ческидов, г. Усть-Каменогорск; Наташа Кузнецова, г. Мурманск; Женя Гарбер, г. Днепропетровск; Дима Тихонов, г. Краснокаменск; Дима Мешков, г. Бердск; Оксана Соболева, г. Харьков; Аня Подшивалова, г. Ленинград; Наташа Кулемина, Архангельская область; Света Шевченко, г. Харьков; Юля Евсеева, г. Красноярск; Оля Шамина, г. Хабаровск; Сергей Яковлев, г. Таллин; Лена Дудышева, Приморский край; Ира Горемыкина, г. Братск.

Конечно, были и программы с ошибками, но это не беда: как сказал Чип, в программировании ошибки легко исправить; надо заставить компьютер работать по этой программе, и сразу будет видно, правильная она или нет.

— А если компьютера нет? — спросил Сережа. — Ведь почти ни у кого из ребят пока нет компьютеров ни в школе, ни дома.

— А тогда можно сыграть в компьютер! Самому сделать все, что написано в твоей программе. Давай возьмем какое-нибудь письмо и попробуем сыграть в компьютер по его программе. А чтобы автор не смущался, мы не будем называть его фамилию. Дай-ка вот это письмо. Так, это пишет Аня Ф., она собирается поступать в кружок программистов, пишет, что наши игры ей помогают учиться. Ну, давай, ты будешь читать ее программу, а я буду играть в компьютер, это мне нетрудно, я ведь и есть мозг компьютера.

— Сначала Аня перечисляет жильцов, это я не буду читать, а вот...

```
Глава № 2. УВИДЕЛА МЫШКА-НОРУШКА ТЕРЕМОК И СТАЛА
ЖИТЬ.
```

Кажется, это правильно?

— Правильно, да не совсем. Зачем же она пишет: «мышка-

норушка», если только что назвала ее «жилец № 1»? С номерами все короче можно записать. Ну, ты дочитай до конца, а я потом все сразу скажу.

— Ладно, — согласился Сережа, — дальше она пишет, как мы с тобой.

```
Глава № 3. Сейчас номер жильца N=1, а потом он будет меняться.
Глава № 4. Вспомните, чему равняется N и к этой цифре прибавьте 1.
```

— А вот тут что-то странное, она забыла, для чего нужно N. Смотри, что она дальше пишет:

```
Глава 5. ПРИШЕЛ ЖИЛЕЦ № 2 (ЗАТЕМ СЛЕДУЮЩИЙ ЖИЛЕЦ).
Глава № 6. СТАЛИ ЖИТЬ ВМЕСТЕ.
```

Так, а теперь она снова вспоминает про N и пишет:

```
Глава № 7. Если N = 6, то переходите к главе № 10, иначе читайте дальше.
Глава № 8. ЖИВУТ ДРУЖНО ВМЕСТЕ. Дальше опять, как у нас.
Глава № 9. Возвращайтесь к главе № 4 и читайте следующие за ней главы.
Глава № 10. ПРИШЕЛ МЕДВЕДЬ И РАЗДАВИЛ ТЕРЕМОК.
Глава №11. Конец сказки.
```

— Ну, теперь давай играть в компьютер, — сказал Чип, — вот дошел я до главы № 7, N у меня сейчас 2, потому что в третьей главе я запомнил, что N = 1, а в четвертой прибавил еще 1. В главе № 7 я читаю дальше, так как N у меня 2, а не 6. А в главе № 9 я возвращаюсь к главе 4, прибавлю к N единицу, иду к главе 5. А вот тут внимание! Написано, что пришел жилец № 2, а ведь надо уже № 3.

— Да, — вступился Сережа. — но она написала в скобках «затем следующий жилец».

— Что значит следующий? Следующий за № 2 — это № 3, но потом ведь нужен будет № 4 и так далее. Непонятно написано, и компьютер ошибется, а виновата будет Аня. Ей надо было просто написать:

```
Глава № 5. ПРИШЕЛ ЖИЛЕЦ № N.
```

Кстати, можно было и про мышку не писать вначале, а вместо

этого в главе № 3 написать, что сейчас N = 0, вместо 1. Видишь, тогда счет начнется с единицы, и в главе № 5 придет мышка. А в главе 2 можно написать: «СТОИТ ТЕРЕМ-ТЕРЕМОК, ОН НИ НИЗОК, НИ ВЫСОК», или как там в этой сказке? Но это уже необязательно. Чтобы Анина программа правильно работала, достаточно исправить только главу № 5, как я сказал.

— А все-таки Аня молодец, — сказал Сережа, — почти правильно написала, а ей никто не помогал. Чип, как ты думаешь, почему нам написали так много программ про теремок, а про красную шапочку так мало?

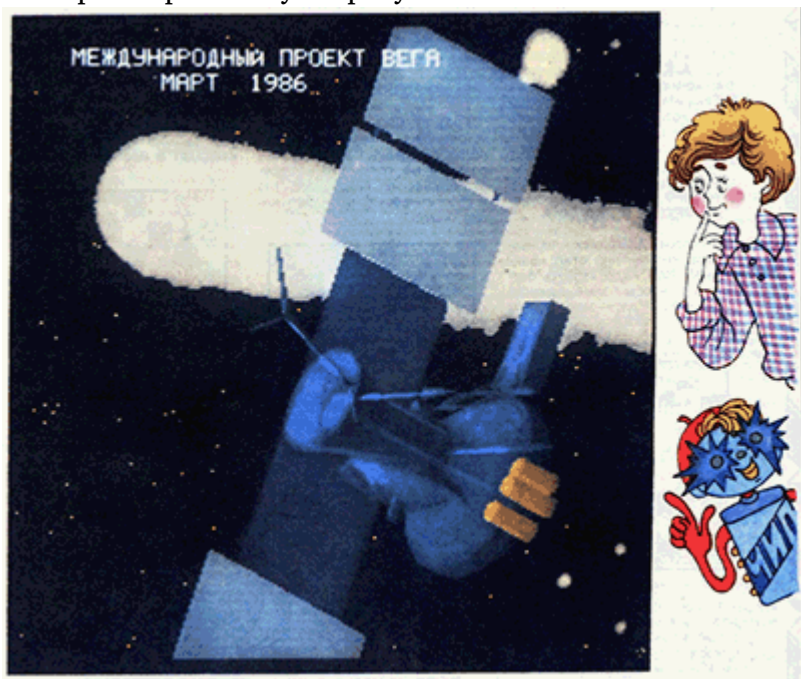
— Наверно, мы слишком трудную сказку предложили, — ответил Чип. — Сказка «Теремок» похожа на «Колобок» и на «Репку», а «Красная Шапочка» совсем другая. Но, чтобы ребята не думали, что сочинить сказку-программу «Красная Шапочка» так трудно, я попробую сейчас это сделать. Так... Готово. Но сначала нужны две подпрограммы:

```
Подпрограмма «У ДВЕРЕЙ (гость)».
1. КТО ТАМ?
2. ЭТО Я, К. Ш., ПРИНЕСЛА ГОРШОЧЕК МАСЛА.
3. ДЕРНИ, ДЕТКА, ЗА ВЕРЕВОЧКУ, ДВЕРЬ И ОТКРОЕТСЯ.
4. Если гость — волк, то он съедает бабушку.
5. Возврат.

Подпрограмма «ДИАЛОГ (глаза, видеть)»
1. БАБУШКА, ЗАЧЕМ У ТЕБЯ ТАКИЕ БОЛЬШИЕ ГЛАЗА?
2. А ЭТО, ЧТОБЫ ЛУЧШЕ ТЕБЯ ВИДЕТЬ.
3. Возврат.

Сказка-программа «КРАСНАЯ ШАПОЧКА»
1. Прочтите сказку Перро «Красная Шапочка» до того места, как волк стучится в дверь бабушкиного домика.
2. У ДВЕРЕЙ (волк).
3. НИЧЕГО НЕ ПОДОЗРЕВАЮЩАЯ К. Ш. СТУЧИТСЯ В БАБУШКИНУ ДВЕРЬ.
4. У ДВЕРЕЙ (К. Ш.).
5. ВОЛК ЛЕЖИТ В КРОВАТИ, ПЕРЕОДЕТЫЙ БАБУШКОЙ.
6. ДИАЛОГ (глаза, видеть).
7. ДИАЛОГ (уши, слышать).
8. ДИАЛОГ (зубы, съесть).
9. ВОЛК СЪЕДАЕТ К. Ш.
10. Если хотите, придумайте счастливый конец сами.
```

11. Конец.

— Молодец, Чип, здорово придумал, но это действительно сложная сказка. Давай теперь на прощание подарим что-нибудь всем ребятам, ведь наступает Новый год!

— У меня как раз есть красивая картинка, ее нарисовал компьютер. Это советская космическая станция «ВЕГА», которая пролетела мимо Венеры и встретилась с кометой Галлея. Компьютер изобразил эту встречу.



С Новым годом, ребята!

Введение

Дорогие ребята! Те из вас, кто выписывал «Пионер" в прошлом году, уже знакомы с Чипом и Сережей. Для наших новых читателей скажем, что Сережа — это обыкновенный мальчик, а его друг Чип — маленький волшебный человечек, который выпрыгнул как-то раз из Сережиного калькулятора. Чип научил Сережу многим интересным вещам: как считать на пальцах до 1000, как помочь царевичу выбрать невесту, как вытащить из лужи барона Мюнхгаузена, как провести конкурс поющих поросят... Да всего и не упомнишь! Эти игры помогают понять основы информатики — новой науки, которая нужна тем, что хочет управлять компьютерами. А так как это придется делать всем в следующем веке, то лучше начать сейчас. Поэтому Чип и Сережа приглашают вас играть с ними вместе.

Сказки восточного базара

— Расскажи мне, пожалуйста, сказку, — попросил Сережа Чипа, — только пострашней и потаинственней, вроде арабских сказок.

— Арабские сказки, говоришь? — Чип задумался. — Что же, могу и арабскую сказку, только сказка эта будет не простая, а рекурсивная, и рассказывать ее будет бродячий дервиш на восточном базаре, чтобы заработать себе горсть фиников и глоток воды.

— А что такое рекурсивная сказка, это вроде рекурсивной программы? — спросил Сережа.

— Это сказка, которая внутри себя содержит такую же сказку.

— Как это может быть?

— Вот сейчас увидишь. Итак...

РЕКУРСИВНАЯ АРАБСКАЯ СКАЗКА

В сказке три персонажа: **СТРАННИК**, **ВОЛШЕБНИК** и **ЗВЕРЬ**.

Если эту сказку рассказывает дервиш,

То СТРАННИК — это мулла, ВОЛШЕБНИК — это джинн[6], а ЗВЕРЬ — это лев.

Если же рассказывает мулла.

То СТРАННИК — это дервиш. ВОЛШЕБНИК — это ифрит. а ЗВЕРЬ — это носорог.

Однажды СТРАННИК ехал на осле по пустыне. Солнце уже высоко поднялось над барханами, и от жары у СТРАННИКА разыгралась великая жажда. «О Аллах! — воскликнул он. — Я отдал бы жизнь за кувшин воды». Эти слова услышал пролетавший мимо ВОЛШЕБНИК. Тут же перед СТРАННИКОМ появился кувшин воды и свирепый ЗВЕРЬ. Великий страх овладел СТРАННИКОМ, колени его ослабели, и, упав ниц на горячий песок, он взмолился: «О могучий зверь! Подожди, не терзай меня, выслушай сначала мою сказку». «Так и быть, — согласился ВОЛШЕБНИК в образе **ЗВЕРЯ**, — можешь рассказать мне одну сказку». «Одну, о мудрый ЗВЕРЬ, только одну!» И СТРАННИК рассказал РЕКУРСИВНУЮ АРАБСКУЮ СКАЗКУ.

ВОЗВРАТ

— Возврат? — переспросил Сережа. — Это куда возврат, в начало, что ли?

— Да нет, — Чип недовольно поморщился. — Ты что, забыл, что такое возврат из подпрограммы? Возврат — это значит конец **этой** подпрограммы и продолжение той, которая работала раньше, пока не началась эта. А в нашем случае это значит, что очередной сказочник замолчит, а тот, кто про него рассказывал, продолжит дальше.

— Погоди, погоди, тут какой-то подвох, — задумчиво сказал Сережа. — Про кого же в сказке речь? Ты сказал, что эту сказку рассказывает дервиш, значит, речь идет про муллу, джинна и льва. Но дальше в этой сказке странник, то есть мулла, рассказывает ту же самую сказку. Про кого же он рассказывает?

— Но в начале сказки ведь было ясно сказано, что если мулла рассказывает эту сказку, то речь идет про дервиша, ифрита и носорога.

— Ах да! Значит, если дервиш, то про муллу, а если мулла, то про дервиша. Ловко! Мулла расскажет льву про дервиша, ифрита и носорога, и в **этой** сказке дервиш уже будет морочить голову носорогу про муллу, джинна и льва. Ну и чем же кончится дело?

— А как ты сам думаешь?

— Я думаю, что у кого-то из волшебников лопнет терпение, и он сожрет странника, не дожидаясь конца сказки. А может быть, у волшебника пройдет гнев и он помилует хитрого сказочника?

— Мне больше нравится второй вариант, но в этой сказке так не произойдет. Мулла и дервиш будут вечно рассказывать друг про друга. Правда, есть одна лазейка из этого бесконечного цикла. Я ведь еще до сказки тебе сказал, что самый первый дервиш, тот, что на базаре, рассказывает сказку, чтобы заработать себе пропитание. Значит, рассказывать он будет до конца дня, не дольше? Потом либо его накормят, либо слушатели пойдут спать. Поэтому мы должны вставить первой строчкой рекурсивной сказки условие:

```
ЕСЛИ кончился базарный день или первого дервиша накормили,
ТО ВОЗВРАТ.
```

— И чем же это поможет?

— А вот смотри. Первый дервиш рассказал рекурсивную сказку до того места, как начал мулла. Дальше снова идет рекурсивная сказка, в начале которой мы вставили условие. Если это условие будет выполнено, то есть если первого дервиша накормили, то мулла кончит сказку, едва начав. А дальше в рекурсивной сказке написано ВОЗВРАТ. Это значит, что и первый дервиш замолчит, схватит свои финики и воду и уйдет.

— А если первого дервиша не накормят к тому моменту, как мулла начнет свою сказку?

— Ну что ж, тогда он ее расскажет до того момента, как второй дервиш начнет свою сказку, а тогда второй дервиш снова проверит, не накормили ли первого. Если да, то он замолчит, а тогда замолчит и мулла, а тогда замолчит и первый дервиш, схватит свои финики и воду и побежит есть.

— Кажется, я начинаю понимать, — протянул Сережа, — каждый очередной рассказчик будет проверять, как там дела на базаре у первого дервиша, и как только того накормят, они все по очереди замолчат. Вот теперь у сказки есть конец, только он так хитро запрятан, что и не найдешь сразу. Я понял, что мне это напоминает. Есть такая пословица: «Иван кивает на Петра, а Петр на Ивана».

— Правильно, — сказал Чип. — Это как раз рекурсивная пословица. Давай дадим задание ребятам, чтобы они записали ее в виде рекурсивной программы. А на конверте пусть напишут: ИВАН И ПЕТР. Лучшую программу мы напечатаем.

А вот имена ребят, которые составили самые интересные программы к считалочкам: Марат КАЛТОВ, г. Гусь-Хрустальный; Лена СУПРОНЕНКО, Приморский край; Вера ГУРЕВИЧ, г. Днепропетровск.

Секрет «Числовых прыгалок» разгадали Катя БАЙМЕТОВА, п. Майский, Бурят. АССР и Оля МАЛАШИНА, г. Новосибирск.

«Красная шапочка» оказалась сложной сказкой для наших

читателей. Но ближе всего к истине были программы Дины КОНОВАЛОВОЙ, г. Владивосток: Маши КВАНТАЛИАНИ, г. Тбилиси; Нади ЧУБЛОВОЙ, г. Тюмень; Маши ШКОЛЬНИК, г. Москва.

МОЛОДЦЫ!

Роза для королевы

— Знаешь, Чип, — как-то раз признался Сережа, — вот мы с тобой уже целый год играем, а мне иногда кажется, что я ничему не научился. То есть я многое узнал, но не знаю, как все это применить.

— Ну и что же тут страшного? Мы же не в школе, и отметки я тебе ставить не собираюсь. Мы с тобой только играем, а в игре можно и ошибаться, и не знать чего-то. И все-таки ты многому научился, сам того не замечая. Давай проверим: я расскажу тебе маленькую историю, а ты подумай, что из того, что ты узнал за прошлый год, можно применить в этой истории.

«Было раннее утро в стране Чудес. В королевском саду щебетали птицы, приветствуя короля, который вышел на прогулку перед завтраком. Но королю было не до них — его прогулка имела важную цель, и он никак не мог сообразить, как этой цели достичь.

Дело в том, что сегодня был особенный день — день рождения королевы, и предстояло выбрать ей подарок. Больше всего на свете королева любила розы, но жалела срезанные цветы и не позволяла дарить себе больше одной розы. Поэтому нужно было выбрать одну, самую красивую розу во всем саду.

Король с тоской оглядел огромные клумбы, усыпанные отборными цветами. Выбрать самую красивую из двух роз он еще сумел бы, но из тысячи... Если каждую из тысячи роз сравнить с каждой другой, это сколько же будет... тысяча раз по тысяче?..

— Билл, сколько это будет — тысячу раз по тысяче? — спросил он садовника, который вертелся рядом, не зная, как помочь королю.

— Не знаю, ваше величество, а только до завтрака не управимся!

— Что же делать, Билл? Ведь мы, пожалуй, и до завтрашнего завтрака не управимся, если все розы будем со всеми сравнивать. Если бы не королева, я бы сорвал первую попавшуюся, а так — нельзя, сам понимаешь...»

— Ну, как, — Чип прервал свой рассказ и подмигнул Сереже, — можешь помочь королю? Нет? Тогда слушай дальше...

«В королевском саду наступила тяжелая пауза, которую прервал скрипучий голос Гусеницы:

ЕСЛИ розу выбираешь, ту, что **лучше** (всех иных),

ТО ты первую сличаешь с той, что **лучше** (остальных).

С этими таинственными словами она исчезла, прежде чем король и Билл пришли в себя от изумления.

— Как тебе это нравится, Билл? — раздраженно заметил король. — Уже гусеницы начинают давать мне советы. И был бы хоть путный совет, а то — сличи первую с самой лучшей из остальных. Это мы и сами знаем, верно, Билл? А вот как найти эту самую лучшую из остальных, не сказала и уползла.

— Ваше величество, я, конечно, человек темный, но эту Гусеницу я знаю — она слов на ветер зря не бросает. Ей-богу, она нам дельный совет дала, да только как его разгадать — ума не приложу.

— А я говорю, что все это вздор! — Король не на шутку разгневался. — Во-первых...»

— Постой, постой, — закричал Сережа, — я понял, понял! Это же конкурс поросят! Которые песенки поют! А я-то слушаю, уши развесил, думаю, что же это мне напоминает?

— Наконец-то, — насмешливо отозвался Чип, — а то я уж не знал, кого еще тебе на помощь позвать. Давай-ка разберем эту программу до конца, чтобы ты потом не говорил, что ничего не понимаешь.

Значит, ты понял, что Гусеница прочла не простой стишок, а рекурсивную программу, вернее, подпрограмму «**Лучше** (среди кого-то)». И что эта подпрограмма уже встречалась нам раньше, в конкурсе поросят.

— Да, и там это называлось «**Лауреат** (среди поросят)». Там тоже надо было сравнить первого поросенка с **лауреатом** (среди остальных).

Только мы сравнивали пение поросят, а здесь — красоту роз, вот и вся разница.

— Ну, и как же должны король с садовником сравнивать розы по этому алгоритму? Можешь рассказать?

— Попробую... Ну, пусть для начала будет три розы, а не тысяча. Тогда они должны сравнить первую розу с лучшей из двух остальных. Это король умеет делать, так что сразу все получится: он сначала найдет лучшую из двух остальных, а потом сравнит ее с первой. Итого два сравнения.

— Так, ну, а дальше?

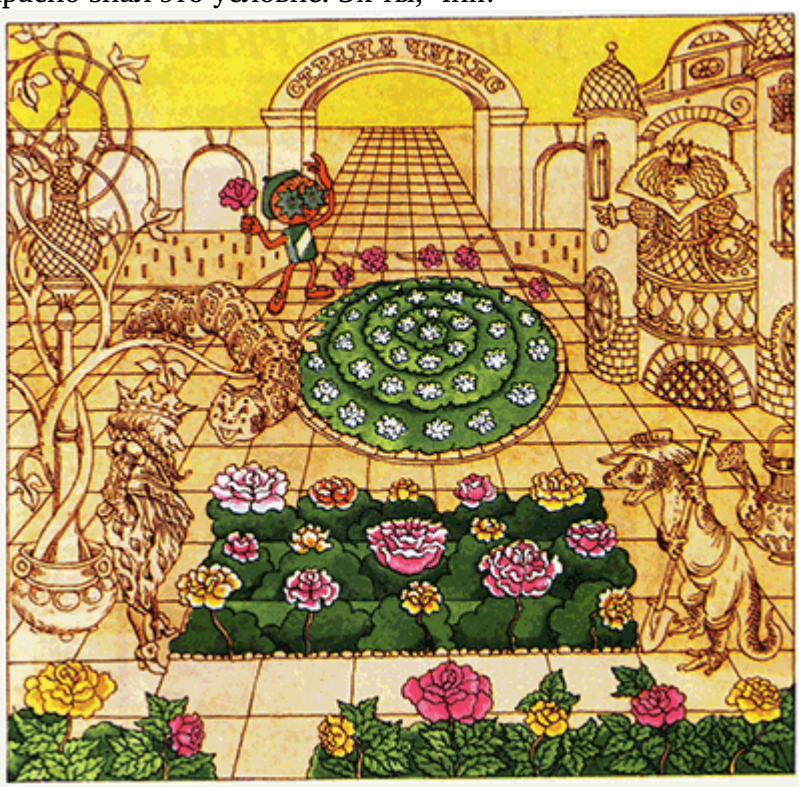
— И дальше так же. Раз мы научились за два сравнения находить лучшую из трех роз, то за три сравнения найдем лучшую из четырех: отложим первую в сторону, как советовала Гусеница, и за два сравнения найдем лучшую из трех остальных, потом сравним ее с первой — и готово дело.

— Ну, а если бы мы не знали заранее, как сравнивать между собой три розы? Ведь именно это остановило короля — он не знал, как сравнивать 999 роз, и потому считал совет Гусеницы вздорным.

— Погоди, мы это, помнится, тоже разбирали с поросятами. Если мы не знаем, как выбирать лучшую из какого-то числа роз, то мы все равно должны откладывать первую в сторону и пытаться выбирать из оставшихся с тем, чтобы потом сравнить лучшую с отложенной. Значит, они будут откладывать 998 роз, пока не останется только две, которые король сможет сравнить. Потому он лучшую из первых двух сравнит с 998-й, лучшую из этих — с 997-й и так далее. За 999 сравнений и 998 откладываний они найдут самую красивую розу в подарок королеве. И вовсе не надо сравнивать тысячу тысяч раз... Чип, что же это получается — они срежут всю клумбу, чтобы выбрать лучшую розу? А как на это посмотрит королева?

— Королева? Гм-гм... — Чип был явно смущен. — Ну, во-первых, они могут не срезать и не откладывать, а, скажем, повязывать на розы номерки, а во-вторых, во-вторых. Гусеница могла и не знать про привычки королевы!

— Чип, во-первых, если король и садовник будут повязывать на розы номерки, они провозятся весь день. А во-вторых, Гусеница, конечно, могла и не знать привычки королевы, но ты-то прекрасно знал это условие. Эх ты, Чип!



ОТ РЕДАКЦИИ:

Ребята, пусть вас не удивляет, что Чип на этот раз ошибся.

Ведь он компьютер, а компьютер думает не так, как человек. Человек, когда решает задачку, все время помнит, для чего нужна эта задача, и ему не всякое решение годится. Ну, а Чип просто не подумал о чувствах королевы, ему важно было выбрать самую лучшую розу — и все.

Попробуйте сами исправить программу Чипа. А чтобы решить ее было проще, считайте, что розы посажены ровными рядами на прямоугольной клумбе.

А как решить ту же задачу, если розы растут не на клумбе, а в один ряд вдоль садовой дорожки?

Ждем от вас два варианта программ. Лучшие работы будут напечатаны. На конверте укажите: «Роза для королевы».

Литературный экскурс в историю счетных машин

— Расскажи мне какую-нибудь историю, пострашней и потаинственней, — попросил Сережа. — Лучше всего про Шерлока Холмса и доктора Уотсона.

— Потаинственней? — Чип задумался и начал зловещим голосом:

«Доктор Уотсон протянул озябшие руки к камину и повторил:

— Нет. Холмс, это не человек, это сам дьявол! Человек не может не оставлять следов и не делать ошибок.

Холмс ничего не ответил. Полузакрыв глаза, он сидел, вытянув длинные ноги, и как будто дремал. Только пальцы, нервно барабанящие по каминной полке, выдавали, что лучший ум Лондона напряженно работает над новой задачей.

Неожиданно он вскочил. Глаза его лихорадочно блестели.

— Едемте, Уотсон, нельзя терять ни минуты!

— Но куда?

— Объясню по дороге. Эй, кеб! Кебмен, нам нужно попасть в Сохо не позднее полуночи. Вот вам соверен, гоните! Уотсон, вам доводилось слышать о леди Лавлейс?

— Ада Лавлейс? Дочь великого Байрона! Уж не хотите ли вы сказать, что она замешана в этой серии дерзких преступлений? Грабительница-аристократка, дочь поэта? Я верно рассуждаю, Холмс?

— О нет, дорогой друг, ваше воображение не в меру разыгралось. Ада Лавлейс замешана в этом деле, но не так, как вы думаете.

— Но как же, Холмс? Я теряюсь, вы говорите загадками.

— Терпение, мой друг, терпение, вот мы и у цели. Этот грязный переулок Сохо отличается от прочих только одним: в нем ровно в полночь, то есть через тридцать пять секунд, произойдет передача краденых драгоценностей... А, вот и они!

Холмс откинулся в глубину кеба и увлек за собой Уотсона. Вынырнув из тумана, на большой скорости приближался черный экипаж. Возница в плаще и маске нахлестывал лошадей. С первым ударом Биг-Бена в доме напротив внезапно отворилось окно, из него вылетел сверток и тяжело звякнул о булыжную мостовую. Из притормозившего экипажа высунулась рука, чтобы подхватить сверток... Но Холмс оказался проворнее.

— Ни с места, Мориарти! — закричал он громовым голосом, осаживая лошадей. Подоспевший Уотсон распахнул дверцу и

приставил револьвер к виску неуловимого профессора Мориарти — некоронованного короля Лондонского преступного мира...

— И все-таки, Холмс, я никак не могу взять в толк, при чем же здесь почтенная леди Лавлейс? — Уотсон с наслаждением отхлебнул горячего кофе, устроился поудобнее у камина и вопросительно поглядел на Шерлока Холмса. — Вы же знали, что это Мориарти, не так ли?

— Напротив, мой друг, я и не подозревал об этом до самой последней минуты. Мои мысли шли по другому пути, на который натолкнули меня вы.

— Я? Но чем же? Я, право, в полнейшем замешательстве.

— А помните, вы давеча сказали, что человек не может не ошибаться? Вы оказались правы, Уотсон, никакой человек, даже пресловутый Мориарти, не смог бы так точно и аккуратно работать, как машина Бэббиджа.

— Машина Бэббиджа? Позвольте, позвольте, это тот чудак, который насмешил все лондонское общество лет тридцать назад? Какая-то дурацкая машина, которая складывала два и два дольше, чем это сделал бы самый тупой сыщик из Скотленд-Ярда? Ах, теперь мне понятно, почему вы вспомнили о леди Лавлейс. Она пыталась поддержать несчастного изобретателя, даже написала трактат о думающих машинах. Но о нем все быстро забыли.

— Леди Лавлейс была умнейшей женщиной, Уотсон, — Холмс помешал кочергой угли и добавил с горькой иронией, — пожалуй, слишком умной для нашего косного Лондона. Она лет на сто опередила свое время, и не ее вина, что ее идеи попали в руки негодяя. У Мориарти хватило воображения понять, какие перспективы открываются перед преступником, который может рассчитывать до секунды все свои действия. Он вложил немалые средства в развитие думающих машин, и результаты мы с вами видели сегодня ночью в Сохо.

— Но все-таки, Холмс, как же вы догадались, где и когда состоится передача драгоценностей?

— Именно безукоризненная логика и точность машины оказались роковыми для Мориарти. Ведь то, что рассчитано одной машиной, может быть точно так же рассчитано и другой? — И Холмс, улыбаясь, постучал себя по лбу черенком трубки».

Чип замолчал и с гордым видом раскланялся. Он был чрезвычайно доволен собой, что случалось каждый раз после литературных упражнений независимо от их результата.

Сережа зааплодировал.

— Здорово ты выдумал про Аду и про какого-то Бобиджа.

— Во-первых, не Бобиджа, а Бэббиджа, а во-вторых, Ада Лавлейс, дочь великого английского поэта Байрона,

действительно написала трактат о думающих машинах Бэббиджа, теперь в ее честь один из лучших языков программирования так и называется: Ада. Пусть мальчишки не важничают: первым в мире программистом была женщина. Правда, ей не удалось проверить, как работают ее программы, потому что первые компьютеры были построены только через сто лет, в середине нашего века.

— Чип, а правда, что преступники могут пользоваться компьютером, обдумывая свои преступления?

— Конечно, могут, только сыщики ведь тоже не дураки — и они пользуются компьютером, чтобы раскрывать преступления. Например, представь себе, что в Скотленд-Ярде стоит компьютер, где имеются сведения обо всех преступниках, а инспектору Лейстреду надо найти среди них профессора Мориарти, скрывающегося под чужим именем. Он знает, что профессор очень худ, лыс, высок ростом и что ему больше 50 лет. Кроме того, профессора знает в лицо Шерлок Холмс, так что сможет его при случае опознать. Как должен Лейстред написать программу для компьютера?

— Ну, пусть компьютер отберет всех преступников с этими приметами и покажет Холмсу их фотографии.

— В памяти компьютера записан рост всех преступников, вес, возраст, и, допустим, он по фотографии сможет различить лысину. Но что такое очень худ и что такое высок ростом, компьютер не понимает. Нет, надо еще объяснить компьютеру эти приметы.

— Ну, про рост можно сказать так — выше среднего роста. Ведь средний рост компьютер сможет вычислить? Пусть отбирает только лысых старше 50 лет выше среднего роста.

— Браво, инспектор, вы делаете успехи! А очень худых как отобрать? Тоже выбирать ниже среднего веса?

— Нет, если отбирать ниже среднего веса, то получатся просто худые, а нам нужны очень худые — чем худее, тем лучше. Стоп! Кажется, понял: пусть найдет из них самого худого и покажет Холмсу. Если это не Мориарти, то его можно вычеркнуть из списка, снова найти самого худого и так далее. Если Мориарти действительно очень худ, то мы скоро до него доберемся — не придется перебирать миллион преступников.

— Что ж, поздравляю тебя с первым детективным алгоритмом, — торжественно сказал Чип и церемонно поклонился Сереже. — Давай теперь запишем его в виде программы.

— Давай, — радостно отозвался Сережа. Никогда еще Чип его так не хвалил, и потому теперь было приятно вдвойне.

Программа «ПОИСК МОРИАРТИ»

1. Взять из памяти приметы очередного преступника.

2. **ЕСЛИ** он худ, **И** лыс, **И** выше среднего роста, **И** ниже среднего веса, **И** старше 50 лет, **ТО:**

занести его в список кандидатов и дать ему по 1 очку за каждый недостающий фунт веса по сравнению со средним весом для его роста.

3. **ЕСЛИ** остались еще преступники в памяти, **ТО**

вернуться к шагу 1.

4. Показать Холмсу кандидата с наибольшим числом очков.

5. **ЕСЛИ** это Мориарти, **ТО:**

поймать его и остановить поиск.

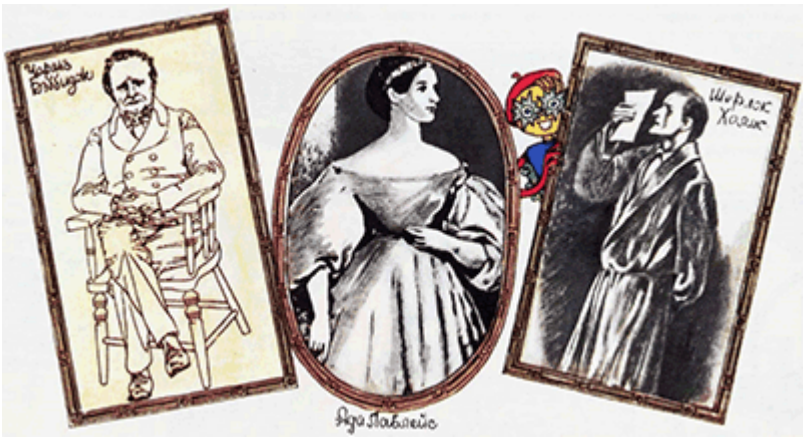
6. **ИНАЧЕ:**

вычеркнуть его из списка кандидатов и вернуться к шагу 4.

7. **ЕСЛИ:** кандидатов больше не осталось, **ТО,**

значит, Мориарти оказался хитрее и надо думать дальше.

8. **КОНЕЦ.**



ОТ РЕДАКЦИИ.

Сенсация! У Шерлока Холмса украли скрипку. Давайте с помощью компьютера поможем знаменитому сыщику ее найти. Приметы скрипки: цвет очень темный, размер меньше среднего, на деке две глубокие царапины. Попробуйте приметы описать так, чтобы вас понял компьютер. Лучшую программу мы опубликуем. Не бойтесь ошибок, главное, чтобы получилось интересно. На конверте напишите: «Поможем Шерлоку Холмсу».

Заклинание для лентяев

Сережа очень любил рыться в старых журналах: «Вокруг света», «Наука и жизнь», «Техника — молодежи»... Там иногда попадались такие интересные факты! Однажды в среду, придя домой после уроков, Сережа вызвал Чипа и они стали вдвоем выискивать интересные сведения.

— Знаешь, Сережа, оказывается, красные чернила бесследно смываются пятипроцентным раствором обыкновенной циклопентанпергидрофенантреновой кислоты. Очень полезная информация для школьника.

— Это что! А вот тут написано, что аборигены острова Мадагаскар умеют жестикулировать носом. Это умение передается по наследству генетическим путем.

— А здесь написано, что некоторые металлы способны направлять и усиливать человеческое биополе. Так, положенный за щеку пятак позволяет передавать и принимать мысли на расстояние до пяти метров. С третьей парты до доски достанет!

— А тут сказано, что в странах Юго-Восточной Азии существует древняя традиция сушить фунчозу на ушах у доверчивых слушателей первого апреля!

Они расхохотались. Оказывается, каждый из них помнил, что сегодня первое апреля,и только и ждал удобного момента, чтобы разыграть другого.

Когда Чип и Сережа вдоволь посмеялись, Сережа спросил:

— Знаешь,Чип, а я все-таки в прошлый раз не совсем понял, зачем ты в программах пишешь **«начало цикла»**, **«конец цикла»**, разве и так неясно?

— Вот, чтобы ты больше не задавал таких вопросов, я прочту тебе одно стихотворение поэта, пожелавшего остаться неизвестным, о лентяе, тоже пожелавшем остаться неизвестным. А ты мне скажи, сколько в этом стихотворении циклов.

КАЮЩИЙСЯ ЛЕНТЯЙ

Каждый год он на лень свою злится:
Через год перестану лениться!
Каждый месяц себе он клянется:
Через месяц работа начнется!
Каждый день он себя заклинает:
Завтра новую жизнь начинаю!
Каждый час говорит он, вздыхая:
Все, последний часок отдыхаю!
Исправляется с каждой минутой,
Но работа стоит почему-то.

— Ну что ж, — спокойно сказал Сережа, делая вид, что не замечает ехидных намеков Чипа, — в этом стихотворении пять циклов: по годам, по месяцам, по дням, по часам и по минутам. Да, если не знать, что из чего состоит и что за чем следует, то можно и запутаться. А как бы ты написал это стихотворение в виде программы?

Чип только этого и ждал. Вот как он написал.

```
Начало 1-го цикла:
ПОВТОРЯЙ каждый год, ПОКА лентяй не исправится:
«Каждый год он на лень свою злится:
Через год перестану лениться!»
   Начало 2-го цикла:
   ПОВТОРЯЙ каждый месяц:
   «Каждый месяц себе он клянется:
   Через месяц работа начнется!»
      Начало 3-го цикла:
      ПОВТОРЯЙ каждый день:
      «Каждый день он себя заклинает:
      Завтра новую жизнь начинаю!»
         Начало 4-го цикла:
         ПОВТОРЯЙ каждый час:
         «Каждый час говорит он, вздыхая:
         Все, последний часок отдыхаю!»
```

```
                Начало 5-го цикла:
                ПОВТОРЯЙ каждую минуту:
                «Исправляется с каждой минутой,
                Но работа стоит почему-то».
                Конец 5-го цикла;
            Конец 4-го цикла;
        Конец 3-го цикла;
    Конец 2-го цикла;
Конец 1-го цикла.
```

— Ну как? — гордо спросил Чип. — Теперь все ясно, не то, что раньше!

— Ты, пожалуйста, не сердись, но только мне как раз наоборот раньше все было понятно, а вот теперь... Где начало, где конец, какие-то циклы, ей-богу, без циклов было проще!

— Это для вас, людей, так проще, а для компьютеров без циклов нельзя писать — собьется компьютер. В том и состоит искусство программирования: писать так, чтобы компьютер мог работать без ошибок. Человеческий язык для компьютера полон загадок.

— Вот, смотри, — продолжил Чип, — начало и конец каждого цикла написаны на одном уровне — сразу видно, какой цикл кончился, а какой нет.

— Все равно непонятно, — упорствовал Сережа, — давай разберем, как мы делали раньше, кусочек твоей стихотворной программы. Может, тогда я пойму.

— Давай, — согласился Чип. — Начнем с главного цикла. Мы можем сначала написать только его.

```
Начало 1-го цикла.
ПОВТОРЯЙ каждый год, ПОКА лентяй не исправится:
«Каждый год он на лень свою злится:
Через год перестану лениться!»
Конец 1-го цикла.
```

— Надеюсь, все ясно? В начале года, скажем, 1 января, лентяй вспоминает про свою лень, злится и обещает через год исправиться. Потом на весь год про это забывает. Поскольку так от лени не избавишься, он организует второй цикл и вписывает его внутрь первого.

```
Начало 1-го цикла.
ПОВТОРЯЙ каждый год, ПОКА лентяй не исправится:
«Каждый год он на лень свою злится:
```

```
Через год перестану лениться!»
   Начало 2-го цикла.
   ПОВТОРЯЙ каждый месяц:
   «Каждый месяц себе он клянется:
   Через месяц работа начнется!»
   Конец 2-го цикла.
Конец 1-го цикла.
```

— Теперь он каждое 1 января повторяет первую пару строчек и, кроме того, каждый месяц, скажем, каждое второе число, повторяет вторую строчку.

— А, понял, — подхватил Сережа, — это тоже не помогло, тогда он вписывает внутрь 2-го цикла третий, по дням. Будет еще каждый день перед сном обещать завтра начать новую жизнь. Здорово получается: пару строчек напишешь, а 365 раз будешь программу выполнять. Только все равно это не поможет, я пробовал. Чип, а что ты посоветуешь, как бороться с ленью? Правда, без шуток.

— Я знаю только один способ бороться с ленью: если ты начнешь делать свое дело хорошо, оно станет тебе интересно и ты будешь его делать с охотой. А если ты работаешь из-под палки, только чтобы отделаться, дело так и останется скучным и тебе будет лень. Сделай свою работу интересной, и лень пройдет. Вот мы с тобой играем, и нам не лень составлять программы, а если бы мы учились по учебнику, мы бы тоже, наверное, ленились.

ОТ РЕДАКЦИИ:

Сегодня Чип и Сережа предлагают вам забыть про лень и заняться таким делом. Представьте, когда часы бьют, из них

выглядывают раз в час петух, а раз в час кукушка. А каждые 12 часов они появляются вместе и хвалят друг друга. Как составить программу для таких электронных часов?

Простой вариант программы: одна кукушка каждый час поет и каждые 12 часов танцует.

Лучшие программы мы напечатаем. Ждем ваших писем. На конверте укажите: «Кукушка и петух».

Приключения электронного мальчика с пальчик

— Знаешь, как меня назвала одна девочка в письме? — гордо сказал Чип в первое теплое майское воскресенье, когда они с Сережей устроились у раскрытого окна. — Ни за что не угадаешь! Электронный мальчик с пальчик, вот как.

— И прекрасно, — подхватил Сережа, — расскажи мне сказку про современного электронного мальчика с пальчик, вроде сказок для роботов Лема.

Чип не заставил себя долго упрашивать, видно, маленькому хвастунишке нравилось рассказывать про себя. Он уселся, поджав ножки, на коробке своего калькулятора, поднял с важным видом палец кверху и начал так:

— В некотором вычислительном центре, в некотором машинном зале жил да был маленький Чип. Он был самым младшим в семье. Шестеро его старших братьев и его родители день и ночь гнули спину на своего хозяина — Центрального Процессора. Работа перепадала и Чипу, да какой с маленького прок! Поздно вечером, когда работы было поменьше, родители ворчали: «Одна морока с этими малышами, все приходится за них доделывать, без них было легче».

И вот как-то ночью Чип услышал, что родители сговариваются завести его с братьями в Лес Неправильных Программ да там и оставить. А это был страшный лес, там на каждом шагу подстерегали опасности: то бесконечный цикл закрутит, засосет, и не вырвешься; то Двоичное дерево своими ветками-рогатками запутает, затянет — начнешь их обходить и сгинешь без следа, только тебя и видели.

Но страшнее всего в том лесу был огромный великан Гигабайт. Сидел он в своей Директории, окруженный верными слугами — Дисководами, и поджидал неосторожных путников. Ему сотню бит проглотить, как муху, — даже и не заметит. Вот к нему-то и привели братьев жестокие родители.

Великан был сыт, видно, перехватил сотню-другую килобайт на ужин, поэтому он отложил добычу на завтрак, а пока решил развлечься загадками, чтобы нагулять аппетит.

— Эй, мелюзга, видите эти весы? — Он показал на огромные весы с коромыслом и двумя чашками. — Если догадаетесь, как найти самого легкого, не больше шести раз сравнивая вас друг с другом на этих весах, то этого самого легкого я отпущу.

Братья призадумались. Конечно, самый легкий — маленький

Чип, но это нужно **доказать**, всего шесть раз сравнивая братьев друг с другом. Если бы можно было начать с Чипа, то, конечно, шесть раз сравнив его с остальными, можно доказать, что он самый легкий. Но коварный Гигабайт не станет их слушать, начнет с кого захочет и все шесть раз истратит на то, чтобы проверить, что он не самый легкий.

И вдруг маленький Чип звонко крикнул: «Я знаю, я знаю! Только ты не обманешь, отпустишь самого легкого?»

Великан расхохотался.

— Эту загадку не смогли решить уважаемые процессоры на радиолампах, не смогли решить их заслуженные ученики на полупроводниках, а ты, малявка, и подавно не сможешь!

— Ну, а все-таки, — малыш не унимался, — если решу, то отпустишь?

— Да я вас всех отпущу и в дорогу провожатого дам, только, ха-ха, вернее всего, вам дорога будет в котел.

— Ну смотри, Гигабайт, все слышали твое обещание, так чтобы потом тебе обманщиком не прослыть.

Великан побагровел, топнул ногой и взревел:

— Клянусь великим Винером[{1}], этот мальчишка выведет меня из себя! Я от своих слов не отступаюсь, я за всю жизнь ни бита неправды не сказал. А ну-ка давай выкладывай свой алгоритм, пока я тебя...

— Итак, слушай, — спокойно сказал маленький Чип. — Берешь первого попавшегося и временно объявляешь его **самым легким**, а потом по очереди сравниваешь остальных с **самым легким**. При этом ты после каждого сравнения временно называешь **самым легким** того, кто на этот раз оказался легче.

— Ну-ка, постой, — перебил его великан, — ты мне что-то голову заморочил, я так не понимаю. Давай-ка проверим! — Он схватил двух самых больших братьев и бросил на весы. — Так, этот вроде немного полегче, объявляем его временно самым легким. — Он схватил еще одного из старших братьев. — Ага, этот еще легче, значит, теперь его временно объявляем самым легким. — Он схватил следующего. — А этот тяжелее, значит, самым легким остается предыдущий. — Четвертый раз сравниваем вот с этим, ага, прежний остался самым легким, и... вот ты, мелюзга, оказываешься самым легким. Хм... действительно, за шесть раз нашел. Это, наверно, случайно так получилось — попробуем в другом порядке сравнивать... так... так, опять за шесть раз этот малыш самым легким оказался!

— Ну как, отпускаешь? — весело спросил младший брат.

Великан насупился.

— Я одного обещал отпустить и свое слово сдержу. Иди и

помни мою доброту.

— А мои братья? — закричал возмущенный Чип. — Ты их тоже обещал отпустить: провожатого можешь не давать!

— Это ты случайно задачу решил, — возразил Гигабайт. — Вот докажи, что ты на самом деле умный, — найди еще один такой алгоритм, и я отпущу еще одного брата...

Чип прервал свой рассказ и хитро посмотрел на Сережу.

— Ну и как ты думаешь, удалось электронному мальчику с пальчик спасти всех братьев? Составил он еще 6 алгоритмов, как найти самого легкого из 7, шесть раз сравнивая всех попарно на весах? Подскажу тебе, что братьев можно разбить на группы: по двое, по трое и так далее и сначала находить самого легкого в каждой группе, а потом уже сравнивать их между собой.

— Погоди, — спохватился Сережа, — ты же мне не объяснил, что такое алгоритм. Это что, правило взвешивания?

— В этом случае это правило взвешивания, а вообще — такой порядок действий, который **всегда** приводит к желаемому результату.

ОТ РЕДАКЦИИ.

Ребята, поможем электронному мальчику с пальчик спасти своих братьев. Напишите нам, кто нашел все 6 алгоритмов, как

выбрать самого легкого из семи братьев. На конверте укажите: «Мальчик с пальчик».

Чип и Сережа называют победителей

— Чип, где ты? — Сережа не на шутку обеспокоился: только что его друг был тут, на столе, и вдруг пропал. — Чип!

— Я здесь! — Еле слышный писк донесся из груды писем, которую принесли из «Пионера». — Помоги, тону!

Отдышавшись, спасенный Чип рассказал Сереже вот что. На конкурс поступило две с половиной тысячи писем, а точнее, 2553.

— Чего тут только нет, посмотри, Сережа! И новые сказки, и стихи, и рисунки, и считалки. Почти сто человек решили все три задачи! Но сначала давай разберем ошибки остальных, чтобы они тоже смогли стать победителями на следующем конкурсе.

Итак, задача первая. Как уговорить маму купить жирафа.

Многие не поняли, что надо написать **программу**, и решили, что мы поручили им уговорить маму на самом деле купить жирафа. Некоторые участники нашего конкурса сообщают, что им не хватило на жирафа денег, другие — что жирафов в этом году не завезли в зоомагазин, а третьи жалуются, что у них зимой слишком холодно, вот летом — другое дело: можно и африканских зверей разводить. Кое-кто из ребят не нашел ничего лучшего, чем просто переписать начало разговора про жирафа из «Пионера»». Это уж совсем никуда не годится — думать надо самим!

Решая эту задачу, ребята проявили меньше изобретательности, чем с остальными заданиями. Наверное, они забыли, что это **игра**, что решать задачу нужно весело, а не просто правильно.

Задача вторая — составить программу чистки картошки.

Типичная ошибка — неправильное употребление подпрограммы. Либо некоторые участники нашего конкурса не читали «Пионер» в 1986 году, где объяснялось, что такое подпрограмма, либо они не поняли этого объяснения. Многие решили, что если просто поставить в скобки концы фраз, то получится подпрограмма. Другие поняли так, что слово «ВОЗВРАТ» означает возвращение к началу подпрограммы, а на самом деле это выход из подпрограммы и продолжение той программы, которая обратилась к подпрограмме.

Простой пример — программа

```
ВОСКРЕСЕНЬЕ.
1. Покатайся на коньках.
```

```
2. Сходи в кино на («Чучело»»).
3. Сделай уроки.
4. Ложись спать.
5. КОНЕЦ.

Подпрограмма Сходи в кино на (фильм)
1. Узнай, где идет фильм.
2. Подойди к кинотеатру.
3. Купи билет на фильм.
4. Посмотри фильм.
5. Выйди из кинотеатра.
6. Вернись домой.
7. ВОЗВРАТ.
```

Видите, программа **ВОСКРЕСЕНЬЕ** вызывает подпрограмму **Сходи в кино на** (фильм), причем в этот раз на фильм «Чучело», а в другой раз это может быть «Ну, погоди!». В конце подпрограммы написано «ВОЗВРАТ», но это не возврат к ее началу — ведь тогда придется снова смотреть тот же фильм, потом снова, и так без конца — нет, это возврат к программе ВОСКРЕСЕНЬЕ. Ты вернулся домой из кино и делаешь уроки.

Задание третье — программа **«СКАЗКИ, СТИШКИ, СЧИТАЛКИ»**.

Это задание у большинства ребят получилось лучше всего. Тут и сказки про кота в сапогах, и про золотую рыбку, и стишки про падишаха, и подпрограмма «Кто пасется на лугу (кони, нет, не кони)». ...Но много и ошибок: некоторые просто нумеруют персонажи или главы и думают, что получилась программа, другие делают бесконечный цикл, третьи забывают увеличивать номер в цикле, и так далее.

Общий совет: программу надо обязательно отладить, то есть посмотреть, как она работает. Ведь программа — это приказы для компьютера, а прежде, чем отдавать приказы, надо проверить, можно ли их выполнить и что при этом получится. Надо сыграть в компьютер — попробовать точно выполнить все эти приказы.

— Ну, хватит, Чип, мне уже надоело разбирать ошибки, — вмешался Сережа, — давай, наконец, выберем лучшую работу.

И тут разгорелись споры! Чип и Сережа чуть не поссорились, пока выбрали 15 лучших работ. Вот имена их авторов: Кирилл МИШАЧЕВ из Липецка (12 лет), Наташа САПУНЦОВА из Таганрога (11 лет), Кирилл ТРУБИЦЫН из Зеленограда (11 лет), Игорь ТЫРНОВ из Харькова (13 лет), Ирина КУЗЬМИНА из Саратова (возраст не указан), Алла КУЗЬМЕНКО из Киева (4-й класс), Алексей АДИГЕЕВ из Ростова-на-Дону (возраст не указан),

Галя УХОВА из Троицка (13 лет), Сергей БАРСУКОВ из Краснодара (13 лет), Елена ЛУЗИНА из Москвы (9 лет), Сергей ВАРЖАНСКИЙ из Москвы (6-й класс). (Сергей у нас уже был победителем конкурса «Теремок», но мы неправильно напечатали в журнале его фамилию. Впрочем, Сергей сам виноват, у него такой плохой почерк, что его не может разобрать даже компьютер.) Дима ЛАПЧИК из Омска (10 лет), Олег КИБИРЕВ из Новосибирска (12 лет), Оксана ШАБЕРНЕВА из Казани (возраст не указан), Женя АНДРЮШИН из Курска (12 лет).

Пять первых работ были лучше всех. Кирилл Трубицын пишет, что занимается в кружке юного программиста. Действительно, по его программам видно: он опытный программист, знает язык БЕЙСИК и многие тонкости работы компьютеров. Но Сережа сказал, что изобретательности и фантазии больше в работе Наташи Сапунцовой, которая использовала только то, что было напечатано в «Пионере». Тут Чип обратил внимание на маленькую ошибку в Наташиной программе про жирафа. Она нумерует пункты своей программы и вводит номер N, который увеличивает на 2 при каждом цикле, чтобы ее программа приводила маме все новые доводы. Но Наташа, к сожалению, забыла написать, когда нужно переходить к пункту N, поэтому ее программа будет, как попугай, повторять маме все тот же первый довод: что Наташа берется ухаживать за жирафом сама. Все остальные доводы, и очень остроумные, так и не будут использованы. Чип сказал, что в программировании нет мелочей, — однажды ракета пролетела мимо нужной планеты из-за пропущенной запятой в программе.{2}

Очень интересная работа у Ирины Кузьминой. Она составила

блоксхемы для чистки картошки и для уговоров купить жирафа и заметила, что они совпадают. Потом она составила настоящую программу на БЕЙСИКЕ для сочинения стихов и прислала эти стихи, написанные компьютером:

Лисица бежала по полю.
Проклиная тяжелую долю.
За нею гнались собаки,
Желая с лисицею драки.

Судя по ее работе, Ирина — программист со стажем, но неизвестно, сколько ей лет, она этого не написала. Поэтому Чип и Сережа решили не присуждать ей первого приза.

Остались две работы — Кирилла Мишачева и Игоря Тырнова. Обе — замечательные, без единой ошибки, ясные и, что тоже важно, краткие. Ведь чем короче программа, тем легче в ней разобраться и тем проще ее изменить, если понадобится.

Вот, например, программа Кирилла для чистки картошки.

```
Возьми картошку и ПОЧИСТЬ (картошку),
а теперь ВЫРЕЖЬ ГЛАЗКИ'.
КОНЕЦ.

Подпрограмма ВЫРЕЖЬ ГЛАЗКИ'.
Если глазки вырезаны, то возврат,
иначе вырежь один глазок и ВЫРЕЖЬ ГЛАЗКИ'.

Подпрограмма ПОЧИСТЬ (картошку).
Если картошка почищена, то возврат,
иначе срежь кусочек кожуры и ПОЧИСТЬ (остаток картошки).
```

Прекрасный пример рекурсивной программы! Эта программа особенно понравилась Чипу, который вообще неравнодушен к рекурсивным программам и считает, что за ними будущее.

А Сереже больше понравилась программа Игоря про жирафа. Он прислал даже два варианта программы. Первый такой:

```
1. Если вам уже купили жирафа, то конец,
иначе продолжайте дальше.
2. Сейчас N = 1.
3. Принеси домой N кошек.
4. Если мама тебя отругала или выбросила кошек на улицу, увеличь N в два раза.
```

```
5. Попроси маму купить жирафа.
6. Если мама купила жирафа, то конец,
иначе переходи к строке 3.
```

— Смотри, Чип, как он здорово придумал: даже если жираф в тысячу раз больше кошки, то за десять дней кошек станет столько, что маме придется выбрать из двух зол меньшее и согласиться на жирафа.

— А смотри, что пишет про жирафа Кирилл:

```
Рекурсивная программа ЖИРАФ.
Чуть-чуть подумай.
Если не понял, почему не надо покупать жирафа, то
ЖИРАФ,
а если понял, то КОНЕЦ.
```

По этой программе ты будешь думать, пока не поймешь, почему не надо покупать жирафа.

Они еще долго спорили и пришли к выводу, что обе эти работы заслуживают первого приза, но работа Кирилла чуть лучше, поскольку он свободно пишет рекурсивные программы и двойные циклы (это он сделал в своей сказке про зайца, которого выгнала из избушки лиса).

Поэтому было решено наградить их обоих калькуляторами: Кириллу дарит калькулятор британская фирма «Джеральд компьютерс», а Игорь получит советский калькулятор «Электроника». Вот фотографии победителей.

Кирилл Мишачев. *Игорь Тырнов.*

Остальные призеры награждены почетными грамотами. Ну, а те, кому на этот раз не удалось получить первый приз, пусть не

вешают носа. Готовьтесь к конкурсу 1987 года!

Как угадать день рождения друга и вывести братьев из Болота Ошибок

Сережа с Чипом болтали о всякой всячине, когда Чип неожиданно спросил:

— Кстати, а когда у тебя день рождения?

— А вот угадай! — поддразнил его Сережа. — Или вычисли, если можешь, ты же компьютер.

— Что ж, попробую, — принял вызов Чип, — только ты мне помогай: после каждого вопроса говори — перелет, недолет или попал.

— Ну, и сколько ты будешь гадать? До вечера-то успеешь?

— Месяц отгадаю после трех вопросов, а день — самое большее после четырех. Спорим? Тогда я начинаю. Июнь?

— Недолет!

— Сентябрь?

— Недолет!

— Ноябрь?

— Недолет!

— Значит, декабрь. Теперь день. 16-е?

— Недолет!

— 24-е?

— Перелет!

— 20-е?

— Недолет!

— 22-е?

— Попал! Это тебе повезло, Чип, а то бы ты дольше гадал.

— Наоборот, мне не повезло. Моим методом, вернее, по моему алгоритму, любой месяц угадывается после трех вопросов, а любой день после четырех, но иногда получается быстрее. Например, если бы ты родился 24 ноября, то число и месяц рождения я угадал бы после пяти вопросов, а не после семи, как сейчас, а если бы ты родился 16 июня — то после двух.

— А если бы я родился на день позже, 23 декабря?

— Тогда бы ты ответил: «Недолет», — на последний вопрос, и я уже точно знал бы, что это 23-го, поскольку 24-го был перелет.

— Это что, все предыдущие ответы помнить? Это надо быть компьютером!

— Нет, все помнить не надо, достаточно держать в голове два последних числа. Алгоритм устроен так, что правильное число всегда находится между этими двумя, а интервал каждый раз уменьшается в два раза. Например, когда я угадывал месяц, я

сначала попробовал июнь — он ровно посередине года, и по твоему ответу я узнал, что твой месяц находится во второй половине. Тогда я снова нашел середину — это сентябрь и по твоему ответу узнал, что твой месяц находится в третьем квартале. Ну, а дальше, проверив ноябрь, я всегда узнал бы месяц по твоему ответу: перелет, недолет или попал.

— Знаешь, месяц меня не так удивляет, ведь их всего 12, а вот что ты один день из 31 узнал за 4 вопроса, это здорово. Значит, ты сначала проверил середину — это 16, а потом остались три вопроса, и ты угадал один из 16 дней. А, пожалуй, не так уж и странно: 16 делим пополам, это 8, еще раз пополам, это 4, а дальше остаются 3 числа, которые угадываются за 1 вопрос. Получается, что с месяцами твой алгоритм работал не в полную силу: если бы их было 16, тебе тоже хватило бы трех вопросов.

— Да, а с днями хватило бы четырех вопросов, если бы в каждом месяце было по 32 дня. Ты, конечно, узнаёшь эти числа — это степени числа 2. Их помнит каждый программист. Если бы компьютер создавал Землю, он бы сделал так, чтобы в году было 16 месяцев, в месяце — 4 недели по 8 дней, в дне — 16 часов, в часе — 64 минуты, а в минуте — 64 секунды. Насколько легче было бы считать нам, компьютерам, ведь мы основаны на двоичной системе.

— А ты напиши в газету: мол, мы, компьютеры, выполняем за людей всю вычислительную работу, а между тем нас до сих пор заставляют пользоваться вашим дурацким календарем и вашими дурацкими часами. Скажи, мол, переведите все на двоичную систему, и мы вам будем быстрее считать.

— Смеешься? Посмотрим, кто будет смеяться лет через 100, может быть, тогда у компьютеров будут и свои газеты, и свои профсоюзы, чтобы заботиться об условиях их труда. Мы не ленимся, но нам обидно, когда нас заставляют впустую работать.

— Чип, милый, ты что, на меня обиделся? Ну извини, я нечаянно. Расскажи мне что-нибудь еще, ты ведь так интересно рассказываешь. — Сережа знал, что его друг больше всего любит, когда его хвалят.

И действительно, Чип дулся недолго, через несколько минут, порозовевший от Сережиных похвал, он уже весело рассказывал очередную историю.

Это была история о том, как маленький электронный мальчик с пальчик Чип, спасший братьев от великана Гигабайта, вел их домой по страшному Лесу Неправильных Программ. Была темная ночь, туман, и братья сбились с дороги. Сами и не заметили, как оказались в Болоте Ошибок — ноги так и вязнут в трясине, и куда

ни ткнись, погружаешься все глубже и глубже.

— Стойте! — закричал братьям маленький Чип. — Мы так все утонем! У меня есть алгоритм, как выбраться из трясины. Я стою на месте, а вы все делаете по три шага, каждый в свою сторону, куда ему кажется лучше. Потом перекликаемся, сравниваем, кто выше всех выбрался, и идем к нему. А чтобы знать, кто выше всех, и заодно чтобы не потеряться, возьмите каждый по веревке, а концы всех веревок будем привязывать к этому колышку. Я останусь возле колышка и по тому, куда идут веревки, буду видеть, кто выше вылез, а кто в трясину попал.

— А если мы все попадем в трясину? — спросил один из братьев.

— Для того я и сказал, чтобы вы дальше трех шагов не отходили. Видишь, дно опускается — сразу назад. Ну а если уж окажется, что во все стороны дно опускается, значит, мы выбрались на бугор, тогда там и останемся до утра...

— Ну и как, — нетерпеливо спросил Сережа, — выбрались они из трясины?

— А тут нечего и гадать, — ответил Чип с гордостью, — если у тебя правильный алгоритм, то рано или поздно ты достигнешь цели.

ОТ РЕДАКЦИИ:

Ребята, а вам такое задание — запишите план мальчика с

пальчик, как выбраться из трясины, в виде программы. На конверте укажите: «Поможем мальчику с пальчик».

Чем кончились мучения Евклида

Наступил август, и родители теперь все чаще говорили Сереже, что пора бы ему повторить программу по математике.

Наконец-то в воскресенье Сережа сел за сложение дробей.

$$\frac{5}{16} + \frac{7}{12} =$$

вывел он в тетрадке и задумался.

Сначала нужно было найти наибольший общий делитель обоих знаменателей, потом разделить произведение знаменателей на этот наибольший общий делитель и найти тем самым наименьшее общее кратное, а потом... Впрочем, потом уже было не так трудно, самое трудное было, конечно, найти наибольший общий делитель.

— Точно так же мучился великий Евклид две тысячи лет тому назад, — сказал насмешливо Чип, когда Сережа пожаловался ему на неподатливость наибольшего общего делителя, — и ты знаешь, чем кончились эти мучения? Он нашел свой знаменитый алгоритм, который навсегда избавил от мучений пытливых искателей наибольшего общего делителя.

— Да я небось не пойму этот алгоритм, мы ведь только с 9-го класса начнем изучать информатику.

— А для таких, как ты, робких искателей Н.О.Д. неизвестный поэт воспел алгоритм Евклида в стихах:

ПОКА меньшее больше нуля, ПОВТОРЯЙ:
Н.О.Д. ему временно ты приравняй,
И большое на меньшее ты раздели,
И остаток ты меньшим теперь объяви,
А большим Н.О.Д. объяви ты теперь,
И к началу вернись, и в Евклида поверь!

— Ну, как я и думал, ничего непонятно, сплошные загадки, — уныло сказал Сережа.

— На то и загадки, чтобы их отгадывать. Ну, подумай сам, тут говорится про какое-то большое, меньшее и про Н.О.Д. Ты, конечно, догадался, Н.О.Д. — это и есть наибольший общий делитель двух чисел, который мы ищем.

— Ну, наверное, большое — это большее из этих двух чисел, а меньшее это меньшее. — Сережа несколько оживился. — Но потом непонятно: эти три числа меняются местами, делятся друг

на друга, я не могу разобраться, что происходит?

— А знаешь, что должен делать программист, столкнувшись с алгоритмом, в котором он не может разобраться?

— Знаю, отложить его в сторону и пойти поиграть в футбол!

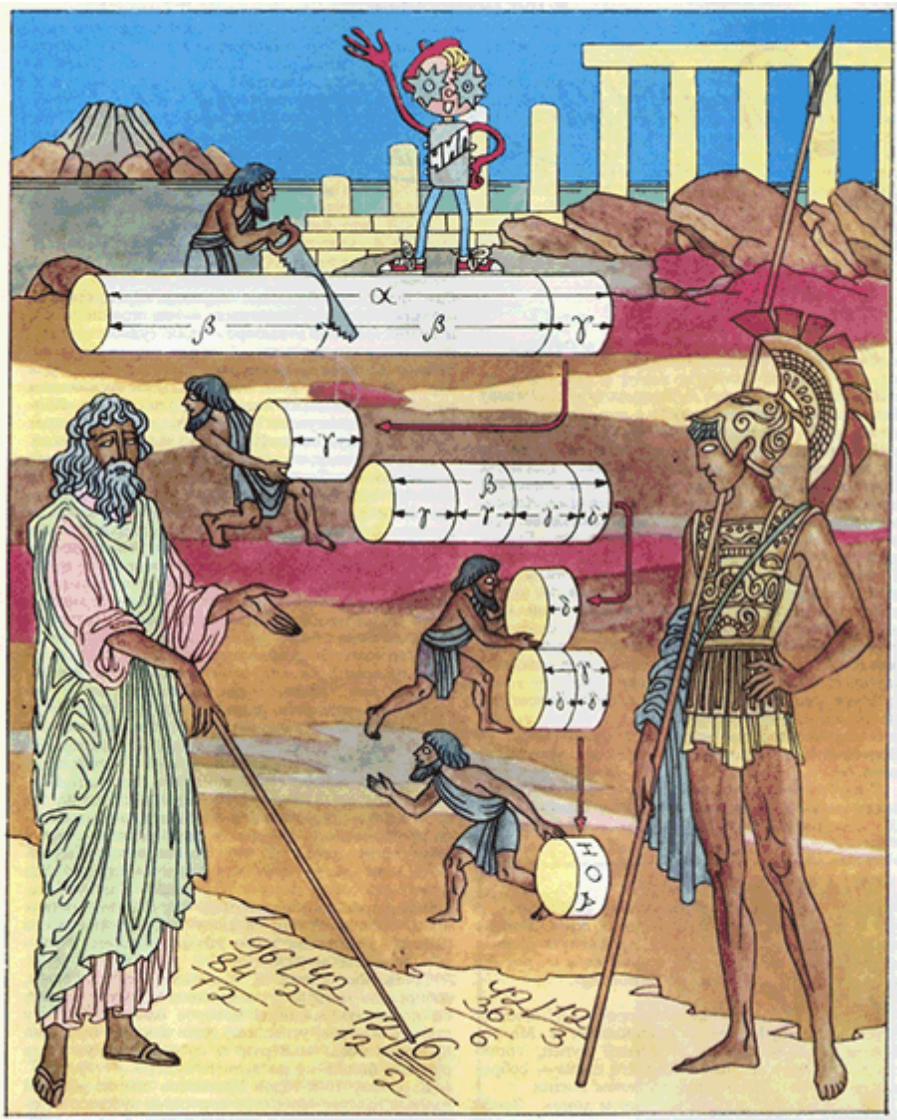
— Я сказал «программист, а не футболист»! Для программиста нет большего удовольствия, чем заставить программу работать. Программист отлаживает программу, то есть проверяет, как она работает, на известных ему примерах. Ну, скажем, ты знаешь, чему равен Н.О.Д. 12 и 30?

— Шести, — ответил Сережа, немного подумав, — 12 — это 2 х 6, а 30 — это 5 х 6.

— Итак, начинаем применять алгоритм Евклида. Малое — это 12, оно больше нуля, значит, повторяем: Н.О.Д. - 12, затем делим 30 на 12, получаем 2 и в остатке 6, значит, объявляем малым 6. Большим объявляем Н.О.Д., то есть 12, и возвращаемся к началу. Малое — это 6, оно больше 0, значит, повторяем снова: Н.О.Д. = 6, делим большое, то есть 12, на малое, то есть на 6, получаем ровно 2. Объявляем малым остаток, то есть малое теперь равно нулю. А большим объявляем Н.О.Д., то есть 6, и возвращаемся к началу. Но теперь малое равно нулю, а значит, повторять ничего не надо, мы уже нашли Н.О.Д. — это 6.

— Что-то не слишком быстро ты нашел ответ, — ехидно заметил Сережа, — я и то меньше думал.

— Долго было объяснять каждое действие, — сердито возразил Чип, — а потом любой алгоритм полезен только в достаточно сложных случаях. Вот посмотрим, как ты найдешь Н.О.Д. 256 и 288 без алгоритма Евклида, и потом сравним, насколько быстрее ты найдешь его с помощью алгоритма.



ОТ РЕДАКЦИИ:

Ребята, а вы не хотите помочь Сереже и тоже выполнить задание Чипа? Решите с помощью алгоритма Евклида пример:

$$\frac{5}{16} + \frac{7}{12} = ?$$

и найдите Н.О.Д. 256 и 288. Ответы пришлите нам.

В № 10 за прошлый год Чип попросил вас составить программу «Приключений в джунглях». Ни одной правильной программы мы не получили. А из всех программ, что вы прислали для рекурсивной пословицы «Иван и Петр», правильная только программа Алисы Левандовской, ученицы 4 «А» класса школы № 45 г. Москвы.

«Я составила рекурсивную программу по образцу рекурсивной арабской сказки. Иван попал в рекурсивную ситуацию.

```
Рекурсивная ситуация.
Если в нее попал Иван, то Игорь — это Петр, Саша — это Иван.
Если в нее попал Петр, то Игорь — это Иван, Саша — это Петр.
Саша кивает на Игоря. Игорь попадает в рекурсивную ситуацию.
Возврат.
```

Есть более короткий вариант этой программы:

```
Рекурсивная ситуация.
Иван кивает на Петра. Петр кивает на Ивана.
Иван попадает в рекурсивную ситуацию.
Возврат».
```

А можно было и так.

```
Рекурсивная подпрограмма КИВАЕТ (Иван Петру).
Иван кивает на Петра;
в ответ КИВАЕТ (Петр Ивану).
ВОЗВРАТ.
```

А чтобы эта программа не зациклилась, то есть не повторяла одно и то же без конца, можно вставить после первой строчки «Иван кивает на Петра» новую команду:

```
СТОП! Их обоих гнать пора!
```

Команда «СТОП», вставленная в любом месте, останавливает всю работу. Хорошо бы и в жизни так можно было прервать любое бесполезное занятие.

Двадцать спичек и монета

Сережа с Чипом играли в увлекательную игру «Двадцать спичек и монета». Кладутся подряд 20 спичек и 21-й — монетка. Дальше играющие по очереди берут спички, рассчитывая так, чтобы последним ходом забрать монетку. Надо только соблюдать два правила: во-первых, монетку нельзя брать первым ходом, а, во-вторых, если противник взял сколько-то спичек, то следующим ходом ты не можешь взять больше, чем это удвоенное число. Например, если он взял 5 спичек, то ты не можешь взять больше 10.

Сережу этой игре научил Чип и потому все время давал ему первый ход как новичку. И тем не менее каждый раз Сережа проигрывал, хотя он обычно хорошо играл в такие игры. Кое-какие секреты игры он понял: например, что невыгодно брать много спичек первым ходом, а если возьмешь больше 6, то противник берет монетку ответным ходом. Сережа брал по одной спичке, и в ответ Чип брал по одной, иногда по 2. Но выигрывал почему-то все время Чип!

— А знаешь, что делает в таком случае программист? — сказал Чип после 15-го выигрыша подряд. — Если он не может понять, как работает программа с большими числами, он проверяет ее на маленьких.

— Знаю, ты мне это уже говорил, — сердито буркнул Сережа. — Только при чем тут программа? Это же игра, а не алгоритм.

— Это алгоритмическая игра: существует алгоритм выигрыша независимо от хода противника. Но самое удивительное, что выигрывает не тот, кто первым начинает! Так что ты ни в чем не виноват: я заманил тебя в ловушку.

— Как это, начинающий обязательно проигрывает? Так не может быть, у него же лишний ход.

— Ты был бы прав, если бы этим лишним ходом он мог не взять ни одной спички. Тогда он поставил бы противника в свое положение. Но по правилам он обязан в каждый ход сколько-то спичек взять, и как только он это сделает, он уходит от спасительного числа Фибоначчи — 21. А до следующего числа Фибоначчи — тринадцати — ему не дотянуть, тогда противник возьмет монетку ответным ходом.

— Что это еще за числа Фибоначчи?

— О-о, это замечательные числа: 1, 1, 2, 3, 5, 8, 13, 21, 34... Они тянутся до бесконечности, причем каждое следующее получается сложением двух предыдущих. Они так хорошо служат программистам — хотя средневековый итальянский математик

Фибоначчи вряд ли мог на это рассчитывать, — что один неизвестный компьютерный поэт прославил числа Фибоначчи в стихах.

— Интересно было бы послушать, — вежливо сказал Сережа, хорошо зная, что Чипа хлебом не корми, а дай почитать свои стихи, которые он обычно приписывал неизвестным авторам.

— Ты знаешь печальную историю о Карле и Кларе?

— Это что, «Карл у Клары украл кораллы, а Клара у Карла украла кларнет»?

— Да, это про них. Грустно, когда друзья ссорятся. Так вот, они решили помириться, и первый шаг сделала Клара.

Однажды Клара подарила
Ему коробку из-под мыла;
Подумав, Карл послал в ответ
Пустой кулек из-под конфет.
Тогда смягчившаяся Клара
Послала 2 воздушных шара,
А Карл послал ей, подобрев,
3 новых карты масти треф.
И с благодарностью от Клары
Пришли 5 варежек без пары;
Как символ дружбы, Карл в ответ
Шлет 8 разных сандалет.
Растрогавшись, послала Клара
13 труб для самовара,
И, прослезившись, Карл послал
21 коленный вал...

Говорят, они стали добрыми друзьями и все время шлют друг другу подарки. И каждый из них, не желая уступить другому в щедрости, прибавляет к своему последнему числу подарков последнее число подарков, полученных от друга. Так и получаются числа Фибоначчи.

— Замечательное стихотворение, Чип, но я все-таки не понимаю, как ты меня обыгрывал с помощью этих чисел Фибоначчи.

— Предположим для начала, что у нас 2 спички и монетка. Тогда начинающий проигрывает, согласен? Он не может сразу взять монетку, и ему приходится брать 1 спичку, после чего противник одним ходом берет оставшуюся спичку вместе с монеткой. Вот мы и нашли первое спасительное число 3. Если ты оставил противнику это число, он проиграл.

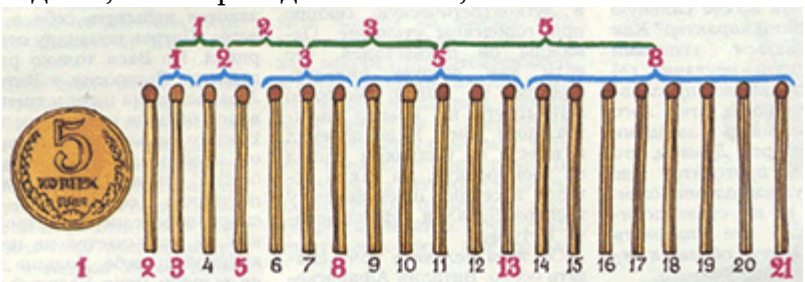
— Ага, а если было 4 предмета — 3 спички и монетка, то можно убрать 1 спичку, и противник проиграет, — подхватил Сережа, — ведь он не сможет взять больше двух спичек.

— Правильно, значит, 4 не спасительное число, если я дам его противнику, он может выиграть. Идем дальше: если начинать с 5, то первым ходом можно взять только одну спичку, иначе противник сможет ответным ходом дотянуться до монетки. Но если ты возьмешь 1 спичку, то ты тоже проиграешь, потому что противник может, взяв 1, оставить тебе проигрышную позицию, которую мы только что разобрали.

— Значит, 5 — спасительное число. Действительно, получаются числа Фибоначчи, но пока я не уловил общего правила — как надо играть, чтобы всегда выигрывать, даже если ты дал противнику спасительное число Фибоначчи. Например,

противник начинает с 8. Мне нужно сделать так, чтобы ему досталось число 5, верно?

— Да, но смотри, ты уже решал эту задачу раньше. От 8 до 5 как раз 3 шага. А для трех предметов ты уже все знаешь: если он берет 1, ты — 2, а если он — 2, ты — 1. Ведь числа Фибоначчи устроены так, что интервалы между соседними числами тоже являются числами Фибоначчи. Ты пытаешься посадить противника на ближайшее число Фибоначчи, а если не можешь, то прибавляешь к этому числу позапрошлое число Фибоначчи. Скажем, от 21 ты должен сажать противника на 13, но этого ты сделать не можешь, значит, прибавляешь к 13 позапрошлое число, то есть 5 (13 + 5 = 18), и стремишься дать противнику число 18. Это просто, так как от 18 до 21 как раз 3 шага — снова число Фибоначчи. Заставил его взять 18 — и тебе осталось 5 шагов до 13, а как проходить 5 шагов, ты тоже знаешь. Вот и все!



— Чип, и все равно это трудно, нужно все время считать и не ошибаться.

— Это трудно вам, людям, а нам нетрудно безошибочно считать. Зато в тех играх, где вариантов слишком много для расчета, например, в шахматах, люди пока обыгрывают компьютеры. Но мы быстро совершенствуемся, так что я не удивлюсь, если чемпионом мира по шахматам 2000 года будет компьютер.

— А чего же вам не хватает сейчас, если вы лучше нас считаете?

— Быстрее — еще не значит лучше. Нам не хватает вашей интуиции, умения без вычислений предвидеть последствия ходов. Интуиция основана на знаниях и опыте, а компьютеры пока не умеют копить знания и набирать опыт. Это будут уметь компьютеры следующего, пятого поколения, которые появятся в начале 90-х годов.

— Чип, а какое задание мы дадим ребятам на этот раз?

— Так как мы сегодня говорили об играх, пусть попробуют придумать сюжет электронной игры, в которую им было бы интересно играть. Не программу, а просто сюжет вроде мультфильма. Автора лучшей игры ждет сюрприз.

ОТ РЕДАКЦИИ.

Ребята, на конверте с этим заданием поставьте девиз: «Сюжет электронной игры».

Совет Чеширского кота

— Чип, что-то давно ты мне сказок не рассказывал, а они у тебя такие интересные получаются!

— О чем же тебе рассказать? Хочешь опять про Алису?

Сережа молча кивнул и приготовился слушать.

— Итак, я расскажу тебе про рыцарский турнир в стране чудес, — начал Чип и тут же сам себя перебил: — А ты знаешь, кстати, что автор «Алисы в стране чудес» и «В Зазеркалье» Льюис Кэрролл на самом деле был Чарлзом Доджсоном, профессиональным математиком, и всерьез занимался теорией турниров?

— Как это теорией турниров? — удивился Сережа. — Турнир — он и есть турнир, сражайся, пока всех не одолеешь, вот и вся теория.

— Вот как? А если твои противники со свежими силами, а ты с каждым боем все больше устаешь? И потом зачем всем рыцарям стоять без дела, пока двое сражаются? Другие пары тоже могут сражаться. Это, брат, целая наука, как провести турнир быстро и справедливо. Вот послушай-ка про один такой турнир в стране чудес.

«Король с утра был в дурном расположении духа. Завтра начинался ежегодный рыцарский турнир, на который съезжались рыцари со всех концов страны. Само по себе это было хорошо, но вот беда, рыцарей в этот раз съехалось уж слишком много — пятьсот двенадцать человек, не считая челяди и оруженосцев. По старым турнирным правилам каждый рыцарь мог сражаться с любыми другими до тех пор, пока не объявится такой рыцарь, вызов которого никто не решится принять. Но сейчас рыцарей было слишком много для таких правил!

— Представляешь, дорогая, — сказал король королеве, — если один рыцарь захочет сразиться со всеми остальными, то это займет 512 дней!

— Пятьсот одиннадцать, — возразила королева, — ведь сам с собой он не станет сражаться.

— Все равно, душечка, это слишком долго, я боюсь, что турнир затянется на весь год, и мы даже не успеем сыграть в крокет, как начнется следующий турнир.

— Ах, какой ужас, — воскликнула королева, — ведь нам придется все это время их еще и кормить! Нет, это невозможно, надо что-то придумать. Может быть, посоветуемся с Чеширским котом? У него иногда бывают прекрасные идеи.

— Представь себе, эта мысль мне уже пришла в голову, и я с

ним посоветовался сегодня утром. Но на этот раз он ничего путного не сказал, если не считать дурацкого стишка. Он прочел его в своей обычной загадочной манере и растаял в воздухе прежде, чем я успел задать хоть один вопрос.

— А ты запомнил этот стишок?

— Изволь, дорогая, я могу его прочесть.

Пятьсот двенадцать рыцарей собрались на турнир,
Чтоб самого достойного узнал волшебный мир.
Они сразились по двое, и до исхода дня
Из каждой пары рыцарей один упал с коня.
И если будут рыцари все так же продолжать,
То сколько дней осталось им друг с другом воевать?

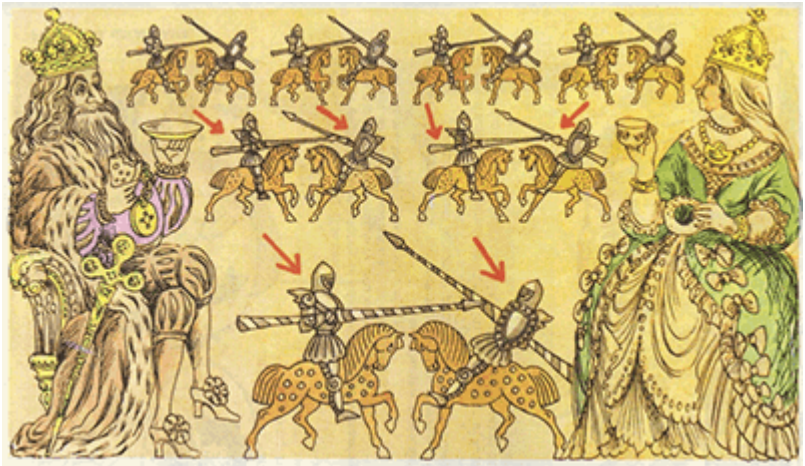
— Знаешь, дорогой, это определенно должно нам помочь, ведь тут тоже 512 рыцарей, как и у нас. Вот только непонятно, как они сражались и сколько времени занял турнир. Может быть, спросим у наших рыцарей? И того, кто отгадает, объявим победителем турнира.

— Ах, что ты, во-первых, им ни за что не отгадать, а во-вторых, они специально сюда приехали, чтобы подраться, и без боя не уедут. Послушай, а что, если спросить Алису?

— Эту скверную девчонку с дурными манерами? — вмешалась возмущенная герцогиня. — Она и крокетного фламинго толком держать не умеет!

— Да, вы правы, милая герцогиня, но нельзя при этом отрицать, что пару раз она нам помогла. Позвать сюда Алису!

— Только и всего? — рассмеялась Алиса, услышав стишок. — Но тут же ясно сказано, как надо сражаться: по двое, все одновременно, а проигравший выбывает. И займет это... дайте подумать...»



— А пока Алиса думает, — продолжал Чип, — мне хотелось бы показать тебе одно письмо. Нас ругает сердитый семиклассник из города Свалява Закарпатской области, который не указал своего имени и обратного адреса, наверное, чего-то испугался. Он пишет: «Ваши темы не заинтересуют даже ученика третьего класса. Нужно сначала выучить язык, какую-нибудь паршивенькую «Рапиру» хотя бы знать досконально, а потом уж разглагольствовать. Моя десятилетняя сестра знает БЕЙСИК, изучает ФОРТРАН 77. Она написала программы для решения уравнений — квадратных, линейных и так далее. Ей смешно читать про «Теремок», про «Красную шапочку», я не говорю уже о себе».

— Прекрасно, — подхватил Сережа, — пусть он нам напишет программу турнира Чеширского кота на любом языке. А чтобы письмо этого сердитого читателя не затерялось в нашей большой почте, на конверте пусть он поставит фразу: **«Чип, это я»** — и обведет ее рамочкой. Впрочем, все ребята тоже могут принять участие в конкурсе, только на конверте пусть просто напишут **«Турнир»**.

ОТ РЕДАКЦИИ:

Ребята! В № 6 за этот год вы увидели фотографии победителей «Конкурса Чипа». А вот как выглядит ПРОГРАММИРУЕМЫЙ НАУЧНЫЙ КАЛЬКУЛЯТОР — приз Кирилла Мишачева.

Такому калькулятору можно задать 40 команд, и он выполнит их все подряд. Например, если вам в школе дали на дом 20 задач на умножение простых дробей, решите на калькуляторе только первую задачу, а с остальными он справится сам, вам останется только вводить в него числа.

Этот калькулятор умеет решать многое из того, что знают старшеклассники и студенты. А младшие школьники могут с ним играть в «Чет — нечет», «20 спичек и монетку» и другие игры. Калькулятор ни разу не ошибется и всегда обыграет вас.

В № 12 за этот год Чип объявит новый БОЛЬШОЙ КОНКУРС. Победителей конкурса ждут призы — КАЛЬКУЛЯТОРЫ.

Вы готовы принять участие в конкурсе?

Случайные числа

— Чип, а не скучно вам, компьютерам, все время иметь дело с числами? Все заранее известно, никаких неожиданностей и случайностей.

— А ты знаешь, что самые сложные задачи, которые решаются на компьютерах — называются они моделирование больших систем, — основаны именно на случайных числах?

— Как это? Я думаю, что чем сложнее система, тем больше приходится считать. А при чем же тут случайные числа?

— Случайные числа не нужны для точных расчетов, но они нужны, чтобы моделировать случайные процессы. А таких процессов великое множество: почти все явления природы и человеческой жизни. Например, голубой цвет неба связан с тем, что лучи других цветов, сильнее рассеиваясь на случайно расположенных молекулах в атмосфере, не достигают наших глаз. Другой пример, из жизни. Как прогнозировать образование очередей или транспортных пробок? Это тоже случайные процессы, поскольку каждый человек принимает решения по одному ему известным соображениям.

— Так что же, получается полный хаос? — Сережа вставил новое слово, которое он недавно где-то услышал. — Может быть так, а может быть иначе? Чем же тут поможет компьютер?

— А компьютер поможет сосчитать вероятности того, будет это так или иначе.

— И как же компьютер это сделает?

— Ну, возьмем простой пример случайного процесса. Когда ты после телефонного разговора вешаешь телефонную трубку, ты обычно не смотришь, как закручен провод, верно? Поэтому после каждого нового разговора он может либо закрутиться еще на один оборот, либо остаться на месте, либо раскрутиться назад на один оборот.

— Он еще может совсем раскрутиться, если папа заметит, что провод слишком сильно закрутился, и не поленится его раскрутить.

— Давай для начала пренебрежем таким редким событием, а остальные три варианта — шаг вперед, шаг назад и шаг на месте — будем считать равновероятными. Это целая наука, как получать на компьютере случайные числа, — для этого лучшие программисты придумывают самые быстрые алгоритмы. Ну, а мы с тобой можем воспользоваться сейчас игральным кубиком. На нем ведь шесть чисел: 1, 2, 3, 4, 5, 6. Давай будем делать шаг назад, если на кубике выпадет 1 или 2, оставаться на месте, если

это 3 или 4, и делать шаг вперед, если это 5 или 6.

— Давай. Ты будешь складывать шаги, а я буду кидать кубик, — оживился Сережа, которому немного наскучили лекции Чипа и захотелось поиграть.

Сережа бросил кубик. Выпало 5, и он крикнул Чипу: «Вперед!» «Один!» — крикнул в ответ Чип.

Сережа кинул кубик еще раз. «Стой!» — крикнул он, потому что выпало 3. «Один!» — повторил Чип предыдущий счет. Дальше дело пошло быстрее: «Вперед!» «Два». «Назад!» «Один». «Стой!» «Один» «Вперед!» «Два». «Вперед!» «Три». «Стой!» «Три». «Вперед!» «Четыре». «Назад!» «Три».

— Ну, что, — сказал Чип,— мы бросили кубик 10 раз и сдвинулись вперед на 3 шага. Значит, провод закрутился бы на 3 оборота. Будем считать, что после каждых 10 звонков папа проверяет провод и раскручивает его. Начнем все сначала.

Они кинули кубик еще 10 раз. Теперь получилось в сумме 2 шага назад, то есть провод закрутился бы на два оборота в другую сторону. Еще и еще кидали они кубик, и получилось, что за 10 бросаний кубика суммарное число шагов бывало в среднем 3 — 4, а направление бывало разным — иногда вперед, а иногда назад. Всего они сделали 10 опытов по 10 бросаний, и в 5 опытах было положительное направление, в 4 — отрицательное, а в одном опыте после всех 10 бросаний кубика получился нуль, так что провод совсем бы не закрутился. Но это, конечно, была редкая удача.

— Видишь, — сказал Чип, — кое-что мы поняли. Ваш телефонный провод будет со временем закручиваться, иногда в

одну, а иногда в другую сторону, но количество оборотов будет расти, так что папе время от времени придется его раскручивать обратно. А если бы мы с тобой бросили кубик миллиард раз вместо ста, то смогли бы точно подсчитать, как величина закрутки зависит от времени и как часто папе придется раскручивать провод. Вот для того, чтобы быстро сделать миллиард таких испытаний, и нужен компьютер. Кстати, в этом простом случае и без компьютера можно предсказать, что величина закрутки провода будет расти, как корень квадратный из времени.

— Давай теперь сделаем что-нибудь смешное, я хочу немного посмеяться, а то надоело с числами возиться.

— Ты любишь играть в чепуху? Это когда несколько человек пишут каждый свой рассказ, меняясь по кругу бумажками после каждого слова и загибая бумажки, чтобы не подглядывать.

— Да, здорово было бы сыграть, только нас всего двое — так неинтересно!

— А вот с помощью случайных чисел можно играть в чепуху даже одному. Сочиняешь шесть разных рассказов, а потом составляешь один, по очереди выдергивая слова из случайно выбранного рассказа. А из какого рассказа выдернуть очередное слово, решаешь, бросая игральный кубик.

Они взялись за дело и скоро сочинили шесть коротких рассказов.

1. Шел снег. Усталый мальчик брел из школы домой насвистывая.

2. Была жара. Веселый воробышек прыгал с ветки на ветку чирикая.

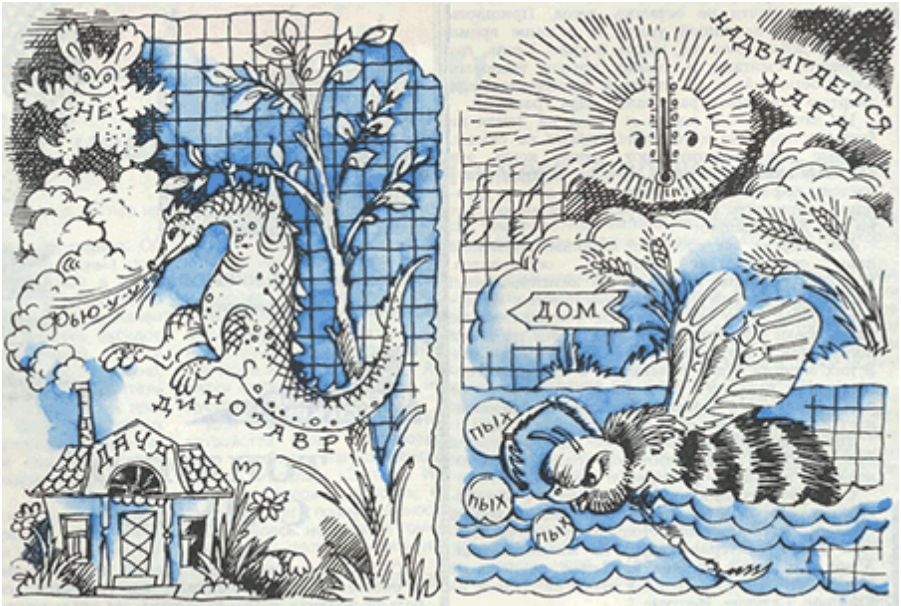
3. Моросил дождь. Огромный кит плыл из моря в океан пыхтя.

4. Лил дождь. Свирепый динозавр выползал из болота рыча.

5. Стоял туман. Сверкающий автомобиль мчался из города на дачу гудя.

6. Надвигалась буря. Озабоченный шмель летел с поля в сад жужжа.

В каждом рассказе было по 10 слов, так что кубик опять пришлось бросать 10 раз. У Чипа получилось: «Стоял снег. Веселый динозавр прыгал с ветки на дачу насвистывая». А у Сережи: «Надвигалась жара. Свирепый шмель плыл с поля домой пыхтя».



Отсмеявшись над своими рассказами, Чип и Сережа придумали такое задание для читателей: сочинить смешную игру, где применялся бы принцип случайных чисел. Лучшая игра будет опубликована в журнале. На конверте укажите: **«Случайные числа»**.

До свиданья, друзья!

Вот и кончается 1987 год. За этот год Сережа узнал много нового и всерьез увлекся информатикой. Он поступил в кружок программистов и научился сам писать программы на языке Бейсик. Теперь настоящий, взрослый компьютер его слушается! Например, попросишь компьютер нарисовать на экране двух рыбок, которые друг за другом гоняются, — и, пожалуйста, бегают по экрану, как живые.

Только, конечно, это не сразу получилось, пришлось помучиться, пока они поплыли. Поначалу рыбки появлялись на новом месте, а на старом не исчезали, так что они не плыли, а плодились. Но потом Сережа догадался, что их надо на старом месте стирать с экрана, а на новом рисовать.

Хотите знать, как пишется такая программа? На экране компьютера каждая строчка пронумерована, а в каждой строчке пронумерована каждая буква. Всего в строчке, скажем, 80 букв, вернее, позиций — мест для буквы.

Скажешь компьютеру: «Напиши букву О на 3-й позиции 5-й строчки», — он ее напишет. А если скажешь: «Напиши знак >, а за ним букву О», — то получится маленькая рыбка.

Рыбка:

```
>O
```

Ну, а щука получится, если нарисовать знак > , два знака равенства, верхнюю кавычку и знак <.

Щука:

```
>=='<
```

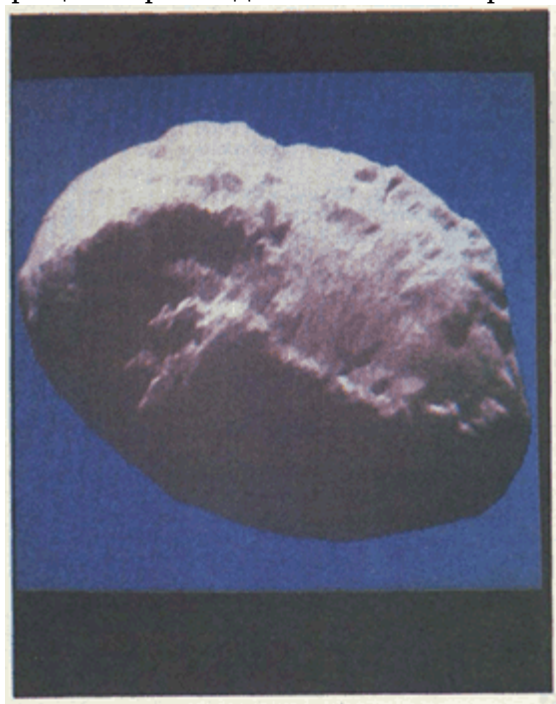
Если нарисовать щуку, начиная, скажем с 3-й позиции, а маленькую рыбку — с 9-й, то получится погоня.

```
  >=='< >O
1234567891011
```

Сережа научил их отталкиваться от стенок, нырять, отдаляться и приближаться друг к другу. Потом он нарисовал голубую воду и зеленые водоросли. Наконец, он сделал так, чтобы щука щелкала пастью.

Жаль, что нельзя показать, какой красивый мультфильм получился!

Зато можно показать кадр из другого мультфильма, сделанного большим компьютером в Институте космических исследований Академии наук СССР. Это Фобос — маленький спутник Марса. Американская космическая станция «Викинг» сделала несколько его фотографий, которые были обработаны на нашем компьютере, так что получилась объемная модель. Ее можно было вращать и разглядывать со всех сторон.



Через 2 года на Марс полетит советская космическая станция, которая со всех сторон облетит Фобос и рассмотрит его вблизи. Этот компьютерный мультфильм нужен для навигации — наша станция будет сверять вид из окна с кадром из мультфильма, чтобы знать, где она находится и куда ей лететь дальше.

— А где же Чип? — спросите вы. У Чипа тоже интересные события. Он как раз и собирается лететь на Марс на космической станции и управлять полетом над Фобосом. Теперь Чип не сможет так часто играть с Сережей. Но они по-прежнему большие друзья и любят в свободный часок поболтать и пофантазировать. А что из этого получится, вы узнаете в новом году.

Нам осталось подвести итоги конкурсов этого года и объявить новый конкурс.

«Поможем Шерлоку Холмсу»

Правильных программ вы прислали много, но 90% из них дословно повторяют то, что было в «Пионере». **Ребята, думайте**

сами! А то в XXI веке вам будет стыдно перед компьютерами.

Вот кто прислал оригинальные решения: Инесса БАГДАСАРЯН из Еревана, Таня АСИЛЬЕВА из Тольятти, Наташа АКИМОВА из Ленинграда, Ирина РОСПАШНЮК из города Фалешты Молдавской ССР, Алексей БЕЛОНОГОВ из Иркутска, А. ЧИЖОВ из Львова, Наталья ЗАВАЛИШИНА из Нижнего Тагила, Д. УГАЙ из поселка имени Морозова Всеволожского района Ленинградской обл., Наташа ВЯТКИНА из г. Сегежа Карельской АССР и Аня КРОЛ из Москвы.

«Роза для Королевы»

На это задание мы не получили ни одной правильной программы, но в трех работах есть интересные идеи. Лина РОСТОВСКАЯ из Москвы, ученица 4-го класса, предложила совершенно оригинальный алгоритм выбора розы с помощью двух палочек. Молодец, Лина! Только в следующий раз попробуй вместо слов «и так будет дальше все время» написать цикл, как мы делали в «Пионере». Аня КРОЛ, которая отличилась и в конкурсе «Шерлока Холмса», тоже прислала интересную работу, написанную белым стихом. Она предлагает разбирать розы на пары, как и Лина, но ее алгоритмы немного сложнее и менее четкие, чем у Лины. Зато она придумала отдельно повязывать лентой непарные розы и потом их тоже сравнивать с лучшей.

Таня САЯПИНА также прислала интересное письмо с правильными идеями и красивым рисунком. Она спрашивает: правда ли, что компьютеры ошибаются?

Вот что по этому поводу думает Чип: «Мы ошибаемся, но гораздо реже, чем люди, и обычно в этом виноваты не мы. Например, если нам написали неверную программу, разве мы виноваты? А еще обиднее, когда прыгает напряжение в сети: считаешь, считаешь, а тут тебе, как обухом по голове, конечно, собьешься. Так что если вы, люди, будете аккуратными, то и мы не будем ошибаться. А пока не слишком нам доверяйте».

Наконец настало время объявить БОЛЬШОЙ КОНКУРС ЧИПА 1987 ГОДА. Как и в прошлом году, победителей ждут КАЛЬКУЛЯТОРЫ.

Условия конкурса:

1. Написать любую программу для калькулятора. Если его нет, то написать программу-сказку, как раньше. Выигрывает тот, у кого самая **интересная** программа.

2. Предложить идею электронной игры. Здесь программа не нужна, будет цениться оригинальность и фантазия.

3. Шутливый конкурс «Кто самый средний?». Назовите год, месяц, число и час своего рождения (не знаете час — придумайте). Выиграет тот, чья дата окажется ближе всех к среднему значению. Вам не надо считать среднее, это мы сделаем сами, вы просто пришлите год, месяц, число и час своего рождения.

На конверте укажите девиз: «БОЛЬШОЙ КОНКУРС ЧИПА 1987 ГОДА». Ответы отправлять не позднее 1 февраля 1988 года.

Как обхитрить Мегафлопа

— Слушай, Сережа, а ты здорово вырос за последний год, — сказал Чип, одновременно для разминки перемножая десятизначные числа. Он вообще никогда не сидел спокойно — говорил фразу и, пока Сережа думал над ответом, успевал решить про себя какую-нибудь задачу. Потом быстро обдумывал Сережины слова, отвечал и снова погружался в расчеты. Однако его собеседник ничего не замечал — так быстро Чип думал.

Чип говорил, что Сережа был для него медленным внешним устройством, а расчет шел в фоновом режиме. То есть на фоне расчета он разговаривал с Сережей, как солист в опере поет на фоне оркестра. Например, ЭВМ может одновременно печатать на машинке и решать сложную задачу. Пока машинка печатает строку, Чип успевает сделать массу вычислений. Потом он прервет решение задачи, пошлет в печать новую строку и возобновит работу с того места, где остановился. Иначе Чипу пришлось бы очень долго, по его понятиям, сидеть без дела, а машины этого не любят. Поэтому в хороших вычислительных машинах могут одновременно решаться десятки проблем, и одна программа не замечает присутствия остальных.

Все это Чип успел рассказать сам себе за ту секунду, которая потребовалась Сереже, чтобы ответить:

— Да. на целых пять сантиметров. А ты почему не растешь? Когда ты станешь таким же большим, как твой папа — Центральный процессор?

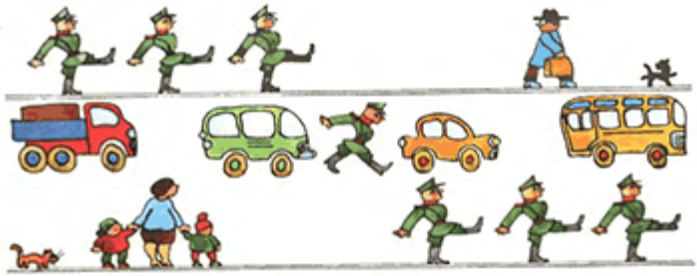
— Мы, чипы, не растем, а сразу рождаемся взрослыми: большими или маленькими, умными или глупыми, быстрыми или медленными. Но бывает так, что несколько медленных чипов оказываются быстрее одного быстрого.

— Десять черепах обгоняют одного зайца?

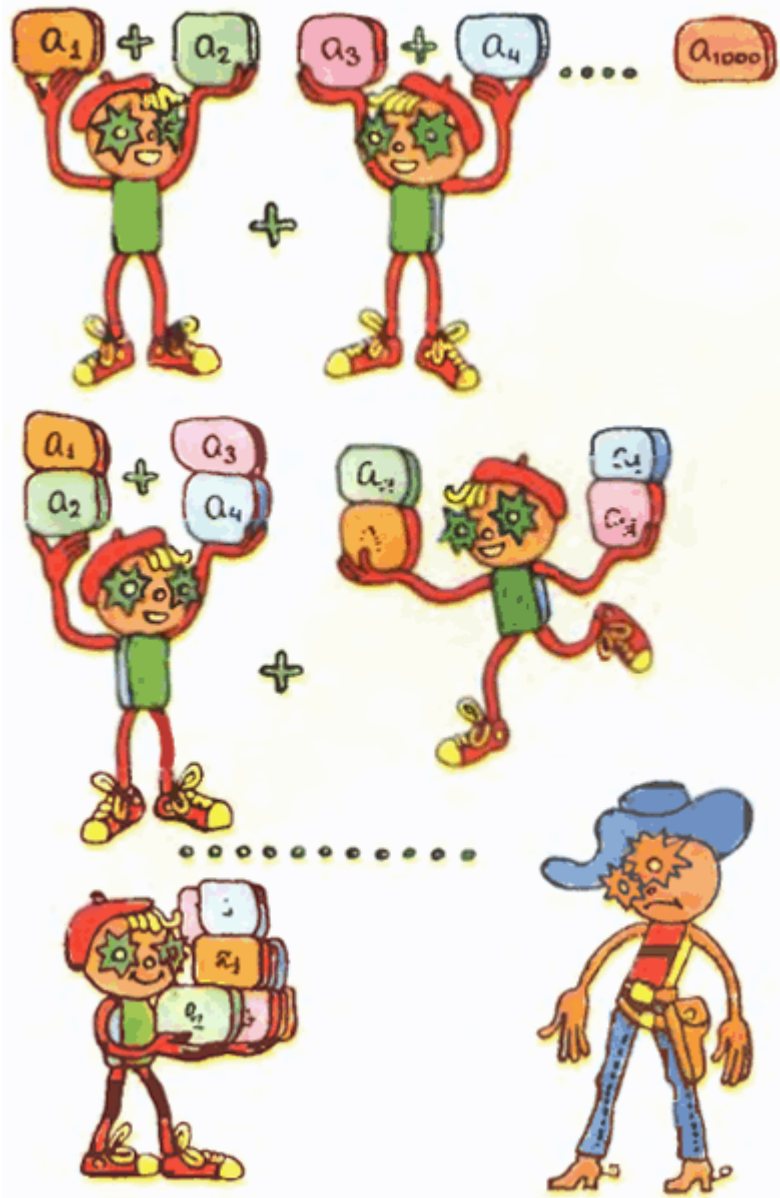
— Почти так, но поскольку машины сами думать не умеют, им должен помочь хороший программист. Вот как-то раз приехал к нам из-за границы нахальный чип Мегафлоп. Был он маленький, гладкий и все ходил и похвалялся: «Эх вы, старикашки! Вам бы на пенсию или в демонтаж! Краска облуплена, час поработаете и перегреваетесь, а уж считаете вы совсем как черепахи. Пока вы два числа сложите, я — сто, пока вы два числа умножите, я — двести. На что вы годны?!» Обидно нам стало, и решили мы пойти к программисту Лене и попросить помощи — неужто и правда нами можно только гвозди забивать?

Леня засмеялся и сказал: «Чипы! Решите-ка такую задачу: по длинной улице идут сплошным потоком машины. Между каждой

парой машин может проскочить один человек. За какое время перейдет на другую сторону улицы рота солдат, если один человек перебегает улицу за десять секунд?». «За тысячу секунд», — хором ответили чипы. «Ну и неверно! Кто вам мешает построить солдат вдоль улицы, и тогда вся рота перейдет за десять секунд.



Так же и из вас, медленных чипов, собирают параллельные машины, а мы, программисты, ломаем потом голову: как распределить работу, чтобы вы все решали одну задачу, но каждый делал свои кусок и не зависел от товарищей. Все крупнейшие современные машины устроены по такому принципу. Предложите вашему Мегафлопу соревноваться. Например, сложить, кто быстрее, тысячу чисел. Он будет суммировать их все подряд, а вы сделайте так, как я нарисовал. Вы вместе за десять шагов сложите всю тысячу потому, что вас много, а я дал вам хороший параллельный алгоритм».



— Погоди-ка, — перебил Чипа Сережа. — Ведь тысяча — это не степень двойки. Разделил пополам — 500, сложил 500 пар, разделил на 250 пар, снова попарно сложил, разделил на 125 пар, сложил — и стоп! Дальше не делится! Не додумал твой Леня алгоритм.

— Молодец, — усмехнулся Чип, — не зря я тебя учил. Что же, в этом случае проще всего добавить три числа, равных нулю.

— А зачем нули добавлять?

— Чтобы всем чипам работа досталась. Сумма не изменится, а слагаемых станет 128, их можно до конца разбивать на пары. А

еще проще, как только количество чисел становится нечетным, к ним добавляется еще одно число, равное нулю.

— Удивительно! Вроде лишнюю работу делаем, а получается быстрее.

— Зато все чипы при деле, в ногу шагают! — Чип подмигнул Сереже и продолжил: - Так мы и сделали, а потом и сами предложили Мегафлопу вторую задачу: найти наибольшее из тысячи чисел. (Кстати, как это сделать за десять шагов?) Потом третью: вывести оценки за четверть для всей школы (для каждого ученика его оценка за четверть по каждому предмету есть среднее арифметическое его оценок, полученных в течение четверти). Как, по-твоему, мы решили эту задачу?

Проиграл Мегафлоп раз, проиграл другой и от стыда сгорел.

Но тут появилась новая проблема! Приехал младший брат Мегафлопа — Гигафлоп. Он работает так быстро, что за ним не угнаться. Давай предложим ребятам придумать такой алгоритм, чтобы все чипы работали одновременно, ни один не стоял без дела.

— А над какой задачей они должны работать? — спросил Сережа.

— И задачу пусть ребята сами придумают. Чипы могут делать что угодно: умножать, делить, сравнивать... из того, что мы научились делать в прошлом году.

— А на конверте пусть ребята напишут **«Сразимся с Гигафлопом»**.

Двоичный поиск и влюбленный принц

— Апчхи! Ну и пылища! — возмутился Чип, как всегда, появляясь из калькулятора. — Ты что, переезжать собрался?

— Нет, у меня генеральная уборка, — вздохнул Сережа. — Ну и скучное же дело! Особенно перебирать шкафы. Я ведь хотел как быстрее: книжки забросил в один, одежду в другой, дверцы прикрыл и пошел гулять. Так бабушка шкафы открыла и заставила перекладывать: чтобы все книги стояли по порядку — по теме, по году издания, по размерам. И с одеждой тоже: сначала маленькая, потом побольше, потом совсем большая — папина... Жуть! И так можно все найти, если постараться: залез, разрыл кучу, нужную вещь вытянул. Пока все книги расставишь, они так надоедят, что больше их читать не захочется.

— Не огорчайся. Даже в такой простой жизненной ситуации можно увидеть интересную задачу и придумать красивый алгоритм. А заодно и понять, почему проще постоянно поддерживать порядок, чем наводить его раз в месяц.

— Какой уж тут алгоритм! Знай рассовывай по шкафам тряпки... Апчхи! Апхчи!

— Замечательный алгоритм быстрой сортировки. Если у тебя есть, например, пятьсот книг, то он ускорит твою работу в тридцать раз. Давай только начнем издалека. Рассмотрим такую задачу: приятель позвал тебя в гости, номер квартиры сказал, а этаж назвать забыл. Предположим, что он живет в очень высоком, скажем, стоэтажном, доме, где на каждом этаже разное число квартир.

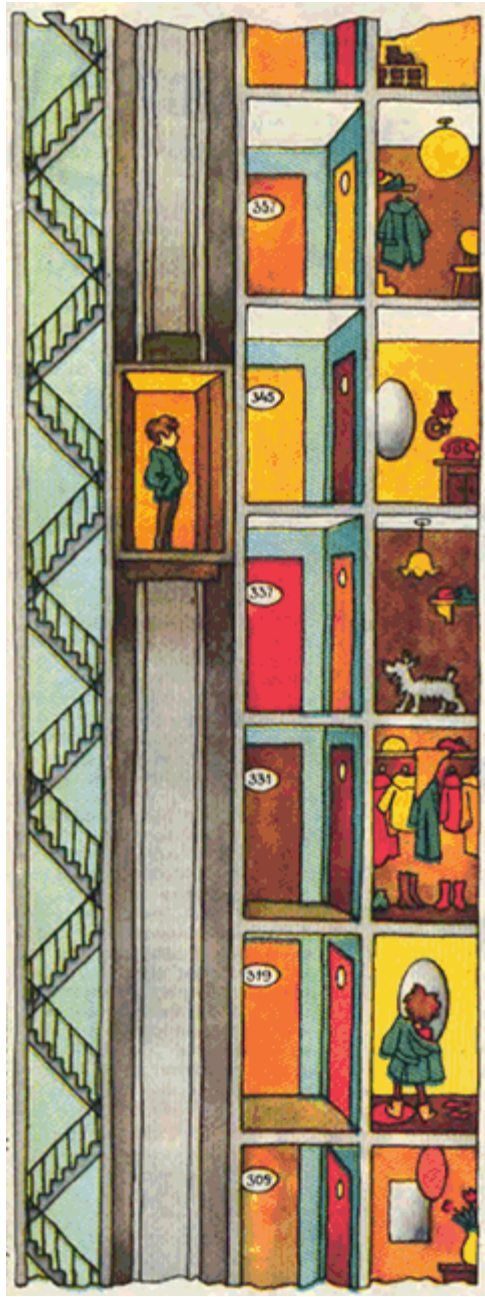
— В Эмпайр-Стейт-Билдинг?

— Пусть там. По числу этажей и количеству почтовых ящиков на первом этаже ничего сказать нельзя. Как бы ты стал искать своего друга?

— По запаху! Раз он меня пригласил, значит, у него пахнет чем-то вкусным, — засмеялся Сережа. — А если говорить серьезно, я бы, наверное, доехал на лифте до верхнего этажа и стал бы спускаться вниз, проверяя номера на дверях.

— И сколько переходов с этажа на этаж тебе бы потребовалось?

— Ну, — задумался Сережа,— если в доме сто этажей и квартира может оказаться на любом, то в худшем случае я осмотрю все сто, а в среднем, наверное, этажей пятьдесят.



— А я утверждаю, что квартиру твоего друга можно найти не более чем за семь проверок. Помнишь стишок про степени двойки?

— Ты хочешь сказать, что два в седьмой — 128? Но при чем тут это?

— А как чипы справились с Мегафлопом, ты помнишь? Для того чтобы нескольким чипам быстро сложить много чисел, эти

числа разбивались на пары, и каждую пару суммировал свой чип, потом суммы снова разбивались на пары и так далее, пока не оставалась сумма всех чисел. Тогда сто чисел мы бы сложили за семь шагов. Если нарисовать график такого суммирования, то получится что-то вроде дерева. А теперь попробуй перевернуть его головой вниз.

— Постой, постой, я, кажется, понял, — обрадовался Сережа. — Мы так определяли день рождения друга. Я должен поехать на пятидесятый этаж, посмотреть там номер квартиры и, если он больше, чем у моего приятеля, поехать вниз, на 25-й, а если меньше, то на 75-й, там опять посмотреть номер квартиры и так далее. Тогда можно за семь шагов найти друга даже в доме со 128 этажами.

— Молодец! — обрадовался Чип. — Вот ты сам придумал алгоритм двоичного поиска. А теперь попробуй написать программу. Давай воспользуемся командой:

```
ПОВТОРЯЙ, ПОКА ВЕРНО УСЛОВИЕ ((БЛОК))
```
[7].

Сережа поднатужился и начал писать, косясь при этом на дверь: вдруг войдет бабушка и задаст ему нагоняй за то, что он отлынивает от уборки?

```
Программа:
1. ВЕРХНИЙ ЭТАЖ = 128.
2. НИЖНИЙ ЭТАЖ = 0.
3. Читай из записной книжки номер квартиры друга.
4. Повторяй, пока ВЕРХНИЙ ЭТАЖ выше НИЖНЕГО.
   ((СРЕДНИЙ ЭТАЖ = (ВЕРХНИЙ + НИЖНИЙ) : 2.
   если ДРУГ живет на СРЕДНЕМ этаже, то КОНЕЦ,
   если ДРУГ живет выше СРЕДНЕГО, то НИЖНИЙ этаж приравниваем к СРЕДНЕМУ,
   иначе ВЕРХНИЙ приравниваем к СРЕДНЕМУ.))
```

— Ишь ты, ни одной ошибки, — сказал Чип, заглянув в листок, — не зря ты учишься в кружке программистов.

— Только какое это имеет отношение к уборке? — отложив ручку, спросил Сережа.

— Самое прямое! — успокоил его Чип. — Давай разберем еще одну задачку. Ты читал сказку Шарля Перро «Золушка»?

— Спрашиваешь!

— Тогда ты, конечно, помнишь, что принц приказал мерить туфельку всем девушкам королевства. Сначала принцессам, потом герцогиням, потом придворным дамам... А теперь

посчитаем. Во Франции порядка миллиона девушек нужного возраста. Если каждая будет мерить туфельку хотя бы в течение минуты, то вся процедура примерки займет около трех лет, и это при условии, что принц будет работать не покладая рук день и ночь. Только сказочная любовь может выдержать подобное испытание. А теперь предположим, что мы с тобой советники короля. Если бы девушек можно было построить, но не по росту, а по размеру ноги, что бы ты предложил принцу?

— Я предложил бы ему объявить по радио, чтобы все девушки измерили свои ноги с точностью до миллиметра и позвонили бы во дворец...

— Это нечестно. Ты же знаешь, что в те сказочные времена не было ни радио, ни телефонов и даже портновский сантиметр был редкостью. Нет, единственный выход - мерить туфельку на ногу.

— Да ведь это та же самая задача! — воскликнул Сережа, немного подумав.

— Я бы...

— Тсс! Пусть лучше ребята пришлют нам программу поиска Золушки. Я думаю, что без нашей помощи принц ее просто не найдет, и сказка кончится плохо!

— Ну, хорошо, а как же построить девушек по размеру туфельки?

— Это могут сделать придворные дамы. Представь, что мы уже построили девушек, и тут пришла еще одна. Тогда нам надо найти двух девушек, стоящих одна за другой, чтобы у одной из них размер ноги был меньше, чем у новенькой, а у другой — больше. Ты понимаешь, как эта задача связана с двумя предыдущими?

— Конечно, нужно только изменить программу. Но ведь придворных дам много. Нельзя ли эту задачу как-нибудь распараллелить?

— Молодец! Не думал я, что ты догадаешься. Но пусть ребята поломают над этой задачей головы сами, а я дам только легкий намек. Предположим, что мы разделили девушек на 2 группы и каждую из них построили. Если ты, используя предыдущие задачи, придумаешь, как соединить эти два строя, ты найдешь самый быстрый метод сортировки, который называется метод «вставок-слияний». А тогда ты не только поможешь королевской женитьбе, но и быстро справишься с генеральной уборкой.

ОТ РЕДАКЦИИ.

Ребята, поможем Сереже решить задачу Чипа? Пришлите нам свои программы, а на конверте напишите: "Где Золушка?» Лучшие программы мы напечатаем.

Бурная жизнь Фатландии

— Чип, давай во что-нибудь поиграем. — предложил Сережа своему другу, — только на этот раз давай играть по-честному)

— А я всегда играю честно! — возмутился Чип. — Жульничать умеет только очень большой компьютер, да и то с помощью программы искусственного интеллекта. А нам, маленьким машинкам, это просто не по силам.

— Ты только не обижайся, пожалуйста, — спохватился Сережа, — я просто имел в виду такую игру, где шансы у противников одинаковые. Помнишь, как мы с тобой играли в спички? Если один из игроков умеет пользоваться числами Фибоначчи, то он всегда выигрывает. И как только секрет разгадан, играть становится неинтересно.

— Понимаю, ты хочешь, чтобы я показал тебе игру без выигрышной стратегии?

— Да. что-нибудь вроде шахмат, шашек или крестиков-ноликов на бесконечном поле, только новенькое.

— Ну, для крестиков-ноликов выигрышная стратегия есть, при правильной игре всегда побеждает крестик. Только алгоритм не такой простой, как в игре со спичками или в игре «орел-решка». Он придуман в Японии, занимает три увесистых тома и состоит в переборе большого числа позиций. Это больше похоже на математическое доказательство, а не на руководство к действию.

— А что за выигрышный алгоритм в игре «орел-решка»?

— Очень простой: не играть вообще.

— Чип, а есть машины, которые умеют играть в крестики-нолики по выигрышному алгоритму?

— Не знаю, но если у какого-нибудь трудолюбивого японца и хватит терпения запихнуть этот многотомный труд в программу, с таким компьютером никто не станет играть. Интересно иметь дело с противником, который играет немного лучше или немного хуже тебя, пускается в авантюры, выдает красивые идеи и вообще думает как-то по-человечески...

— Ну вот, Чип, опять ты сказки рассказываешь! Разве может машина придумать что-нибудь сама? Она знает только то, что есть в программе.

— Не совсем так. Например, для шашек уже существует программа, думающая, «как человек». Ее идея в том, чтобы компьютер все время учился: после каждого хода сравнивал результаты своих прошлых прогнозов с тем, что на самом деле происходит, и в следующий раз делал более точные прогнозы.

Такие прогнозы заменяют ему перебор ходов: он посмотрит несколько ходов вперед, а дальнейшие варианты предугадывает. После каждого хода он прогнозирует все лучше и лучше. Ему ничего не стоит хранить в памяти все сыгранные партии и извлекать из них уроки.

— А как он это делает? Ведь алгоритм прогноза надо было написать заранее?

— Что-то, конечно, надо было написать заранее, в этом и состоит искусство программиста, но можно сделать так, чтобы программа сама себя исправляла.

— Ладно, хватит, Чип, я устал от разговоров. Давай все-таки поиграем.

— Хорошо, давай сыграем в машинную игру под названием «Жизнь». Условия игры такие. Колония бактерий живет на бескрайних просторах Фатландии. Предположим, что эта страна разбита на клетки, как листок из тетради. В каждой клетке только одна бактерия. Соседями одной клетки считаются все клетки, расположенные рядом по горизонтали, вертикали и диагонали. Мерой времени у нас служит смена поколений бактерий, и колония будет жить по таким законам:

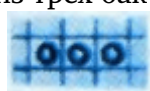
1. Если у клетки меньше двух соседей, то бактерия в ней гибнет от скуки.

2. Если у клетки больше трех соседей, то бактерия в ней гибнет от тесноты.

3. Если у пустой клетки ровно три соседа, то в ней рождается новая жизнь.

Только не применяй эти правила поочередно к каждой клетке, бактерии рождаются и гибнут по общему сигналу.

— Постой, постой, я не успеваю, — перебил его Сережа, — давай посмотрим историю жизни какой-нибудь одной колонии. Пусть у нас будет колония из трех бактерий



— Применяй наше правило. У самых крайних клеток по одному соседу, значит, они погибнут от скуки. У средней клетки два соседа, она не погибнет. Теперь рассмотрим те клетки, которые выше и ниже средней. У них по три соседа. Значит, в каждой из этих клеток в следующем поколении возникает по бактерии.

— Чип, постой. Я не понимаю. Ведь та клетка, которая была сбоку, умерла, значит, у верхней средней клетки нет трех соседей.

— Я же сказал, они умирают и рождаются только по общему сигналу, в Фатландии жесткий порядок. Ты сначала найди все

возможные изменения в своей колонии, а потом уж разом меняй.

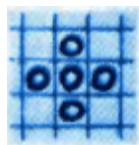
— Понял, понял. Тогда в следующем поколении наша колония станет такой:

а потом опять

и так далее.

— Молодец! А колония

вообще никогда не меняется, но в Фатландии случаются и более невероятные приключения. Поиграй, например, с обыкновенным крестиком

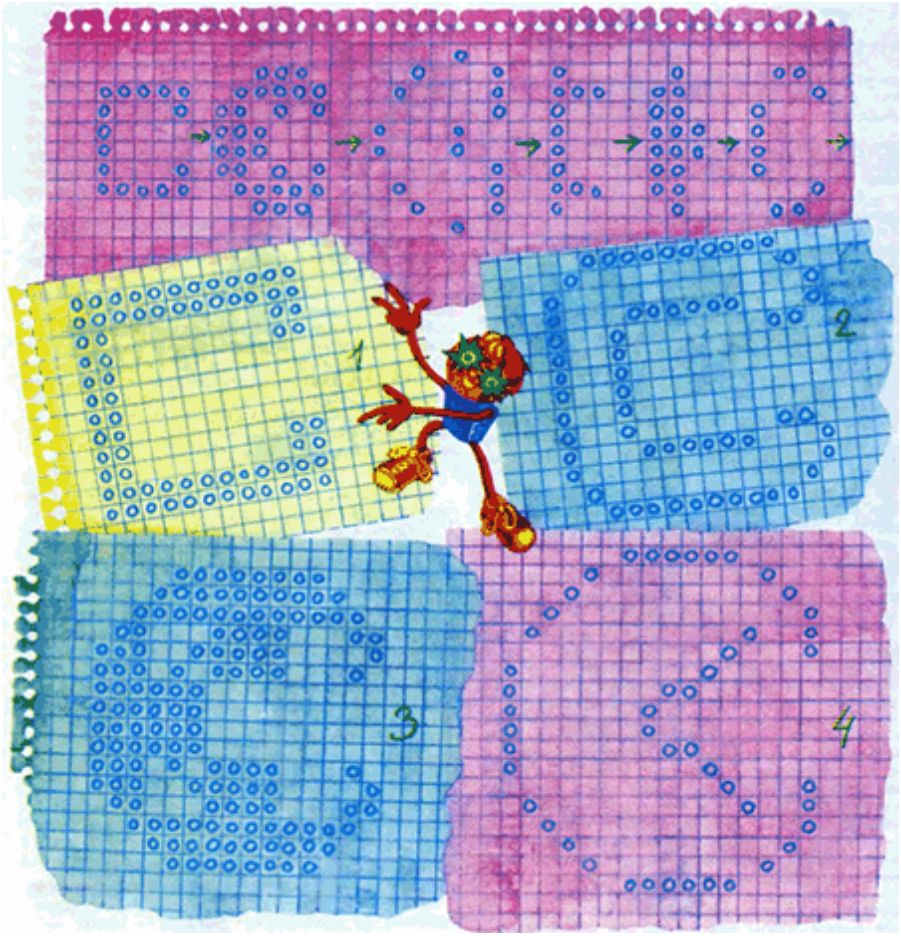


если ты хорошо подумаешь, то поймешь, что через четыре хода он превратится в четыре таких же.

— Чип! А давай напишем программу для «жизни»!

— Давай, только пусть это будет не простой подсчет соседей, а программа, которая ищет игровые ситуации. Например, если колония зациклится (повторит прошлое состояние), то пусть программа на это отреагирует.

Сережа подумал и написал программу. И машина выдала по этой программе вот такую серию картинок.



ОТ РЕДАКЦИИ:

Ребята! А вам такое задание: на листе бумаги в клеточку ноликами нарисуйте первую букву своего имени (такую форму будет иметь ваша колония бактерий) и продолжите историю развития этой колонии. Самую интересную историю мы напечатаем. На конверте поставьте девиз: «Жизнь».

Чип и Сережа читают ваши письма

— Чип! Посмотри, как много у нас новых друзей! — обрадовался Сережа, глядя на стол, заваленный письмами. Он взял наугад несколько конвертов: Мурманск, Одесса, Хабаровск, Ленинград, Киев... — Вот бы съездить ко всем этим ребятам в гости!

— На это тебе не хватило бы даже летних каникул, — засмеялся Чип. — Только на конкурс «Турнир» пришло 602 письма. Кстати, вот задача. Наверняка есть ребята, которые прислали нам по нескольку писем — ответы на различные конкурсы. Как нам найти их ответы в этой груде?

— Разве это задача, — сразу ответил Сережа. — надо рассортировать конверты с фамилиями авторов по алфавиту. А алгоритм быстрой сортировки мы уже знаем, помнишь историю про принца и Золушку?

— Помню, а вот какую программу ты бы написал для нашей задачи?

Сережа немного подумал и написал такую программу:

```
1. Взять мешок, о котором М писем.
2. Рассортировано 0 писем.
3. Повторяй, пока рассортировано меньше, чем М
писем.
4. Возьми из мешка письмо.
5. Вставь в список новое письмо.
6. Конец.

Подпрограмма: «Вставь в список» (новое письмо)
1. Верхний = 0
2. Если рассортировано 0 писем, перейди к строке
7.
3. Нижний = длине списка.
4. Средний = (Верхний + Нижний): 2
5. Если фамилия > Среднего, то
   Нижний = Среднему,
   иначе
   Верхний = Среднему.
6. Если Верхний > Нижнего + 1, переходи к 4-й
строке.
7. Вставить фамилию после Верхнего.
8. Длина списка увеличилась на единицу.
9. Возврат.
```

— Все правильно, — сказал Чип, проверив программу, — только ты ведь не знаешь, сколько в мешке писем. Поэтому лучше будет, если ты напишешь третью строчку программы так:

```
3. Повторяй пока в мешке есть письма:
```

И еще хорошо бы вставить в блок между четвертой и пятой строчками строку: «Прочитать письмо». Кстати, именно таким способом я и воспользовался, когда разбирал нашу почту.

Очень много откликов пришло на задание найти алгоритм Евклида. Правильные ответы прислали: Сергей АЛЕКСЕЕНКО из Хабаровска; Дима АПУХТИН, Челябинск; Люда БРЫНЬ, г Мена; Лена ГЕРАСИМОВА, Одесса; Саша ЗАВОРИН, п. Победа; Олеся КОСТЕНКО, Севастополь; Оля МАЗЕПА, Волжский; Лена МАЙОРОВА, Килок; Саша МАРТЫНЕНКО, Киев; Лена МОСТОЛЬСКАЯ, Москва; Наташа МУСИЕНКО, Похвистнев; Света НЕРЫДАЕВА, Ярославль; Туля ПУЗАТОВА, д. Б. Когульбаева; Таня РАМЕНСКАЯ, с. Рогачево; Аня САМАРОВА. Москва; Алена ТАРАСОВА, Ижевск; Оксана ТЮНИКОВА, Омск; Оля УСАЧЕВА, Донецк; Нина ШЕВЧЕНКО, Иркутск; Юра ЮРЬЕВ, Москва.

Гораздо сложнее оказалось помочь электронному мальчику с пальчик выйти из болота. С этой задачей из 90 претендентов лучше всех справились Алеша АЛЕКСЕЕВ, Уфа; Витя РЫЖАКОВ, Томск и Дима СОСУЛЯКИН, Новокузнецк. Программы которые они предложили, очень похожи и выглядят примерно так.

```
1. Возьми три веревки, привяжи к колышку.
2. Трем братьям отойти в разные стороны.
3. Выбрать верхнего.
4. Всем переместиться к выбранному брату и забить там колышек.
5. Если не вышли из болота — возврат к 1.

Подпрограмма: «Выбрать верхнего».
1. Посмотреть на веревки.
2. Если одна выше остальных — выбрать этого брата.
3. Если все на одном уровне — жди утра.
```

— По-моему, — сказал Сережа, — все трое допустили небольшую ошибку. Чтобы выйти из трясины, нам надо найти не брата, стоящего выше всех, а место, которое выше, чем то, где забит колышек. Возможно, что одна веревка будет выше других,

но все братья будут находиться ниже колышка. Тогда мы не только не выйдем из трясины, но, наоборот, завязнем еще глубже. Программу «Выбери верхнего» я бы написал так:

```
1. Посмотри на веревки.
2. Если есть хоть одна, идущая вверх, то
   выбери самую верхнюю,
   иначе
   жди утра
3. Возврат.
```

— Правильно, Сережа, я бы даже усилил твое утверждение и написал: «Если есть хоть одна, идущая не вниз» — зачем ждать утра, если есть возможность испробовать еще один вариант, оставаясь на том же уровне.

— Посмотри, Чип, вот сюжеты электронных игр, целых семьдесят писем.

— С этим заданием лучше всех справились Никита ЛИТВИНЕНКО, Краснокаменск и Ира РЯБЫХ, п. Ларба. Пятиклассник Никита предложил игру «Леопольд, выходи!» и даже нарисовал картинку. Сюжет игры такой: две мыши, которыми управляют с помощью клавиш компьютера, дразнят кота Леопольда и прыгают через кнопки, лежащие на дороге. Если мышка приземлится на кнопку, раздается громко «Ай!» (современные компьютеры уже умеют разговаривать), и игрок получает штрафное очко. Дорога с кнопками может двигаться быстрее или медленнее. Поскольку мышек две и они прыгают независимо друг от друга, управлять ими будет не просто, такая игра потребует хорошей координации движений и быстрой реакции.

— Чип, а помнишь того сердитого читателя из Свалявы? Он писал, что мы даем слишком простые задачи. Ему предложили более трудное задание и попросили написать на конверте: «Чип, это я».

— Конечно, помню. Письма из Свалявы мы, правда, пока не получили, зато вон лежат семь писем из разных уголков страны с надписью «Чип, это я». Не знаю, что хотели сказать их авторы, но, к сожалению, в этих письмах нет ни слова о сложном задании, просто ответы на конкурс «Турнир». Кстати, с этим конкурсом справились многие ребята. Правильные ответы прислали Дима АНТОНОВ, Пермь. Ярослав ВЕРБЕ, Киев; Паша ДАНИЛОВ, Загорск; Юра ДУКАЧ, Экибастуз; Саша ЕЛЕСЕЕВ, Ростов-на-Дону; Алексей ЕРОХИН, Марганец; Дима ЗАЙЦЕВ, Апатиты; Кирилл ИЛЬИН, Ленинград; Саша КОНЕВ, Курчатов; Сергей

КОРОВЯКОВСКИЙ, Мурманск; Роман КАЛИНА, Оренбург; Таня КРЕСТОВСКИХ, Ухта; Лена МАТАШКОВА, Витебск; Слава НИКОЛАЕВ, с Новомихайловское; Игорь ПЕРМИНОВ. г. Ступино; Алеша МАЛЬКОВ, г. Шимановск; Ольга Л., с. Вихоревка; С. РАССТРИГИН, Рязань; Юра ПЕТРАШОВ, Тольятти; Женя СЕРГУНОВ, Юрга; Дима СУТОРИН, Пермь; Яна СУХОРУКОВА, Красноярск; Оля СИЛИНА, Юрьев-Польский; Ольга ШЕПТАЛИНА, Балашиха; Эльвина ЮНУСОВА, Уфа.

Много программ на бейсике, программ для микрокалькуляторов прислали наши читатели на задание «Турнир». Больше всех постарался Станислав Расстригин: он прислал программы на трех языках — бейсике, алголе и рапире. Игорь Перминов из Ступина на базе нашего задания придумал сюжет электронной игры «Турнир».

— Чип, а давай предложим тем читателям, кто занимается в кружках программирования, усложнять для себя наши задания. Пусть записывают программы на машинных языках или придумывают на основе этих программ электронные игры.

— Конечно, если наши задания кому-то кажутся слишком простыми, их надо использовать как отправную точку для самостоятельной работы.

Образец написания цифр индекса
СЕРЁЖКА
ЛЮБИТ
СЛИВЫ,
ЧИЛИ
СОЛЬ
СЛИВЫ
СЛИВЫ
СЛИВЫ

В поисках выхода

— Чип, а как поживает электронный Мальчик с пальчик? — спросил Сережа. — Расскажи мне про него еще какую-нибудь историю!

— Ладно, — ответил Чип, — только я буду рассказывать, а ты писать программы. Согласен?

— Согласен. — обрадовался Сережа. — Начинай!

— Итак, как-то ночью, когда все инженеры спали, в электрическую сеть прокрался страшный Скачок Напряжения и напал на чипов. Сам Мальчик с пальчик почти не пострадал, его спасли друзья — предохранители, а вот у его старшего брата сгорела вся оперативная память. Жалко стало Мальчику с пальчик братца — теперь тому прямая дорога в демонтаж. И решил он пойти на поклон к великану Гигабайту: ему что проглотить, что отдать пару килобайт — раз плюнуть.

Мама с папой стали Мальчика с пальчик отговаривать:

— Не ходи — и брату не поможешь, и сам пропадешь. Построил себе Гигабайт из каналов и контролеров страшный лабиринт. Кто в этот лабиринт заходит, того великан съедает, а если и не съест тебя Гигабайт, то все равно заблудишься, закружишься в бесконечном цикле, не найдешь обратной дороги.

— Ничего, — отвечает Мальчик с пальчик, — мы с Гигабайтом старые знакомые, один раз я его уговорил, уговорю и в другой.

Взял он узелок, положил туда пару батареек и отправился в путь. Долго ли, коротко плутал по лабиринту, но наконец нашел Гигабайта. Увидел его великан, разинул пасть...»

— Постой, постой. — перебил Чипа Сережа, — это мне напоминает историю про Тесея и Минотавра, я ее читал в «Легендах и мифах Древней Греции». Только ты забыл про Ариадну, которая дала Тесею моток ниток. Он привязал один конец у входа, а потом, когда убил злого Минотавра, по нитке нашел дорогу назад.

— Молодец, — похвалил Чип Сережу, — как раз в нитке все и дело. Герою Тесею, который хорошо умел только махать мечом, нитка действительно была необходима. А наш электронный Мальчик с пальчик выйдет из лабиринта без всякой нитки. Так что это не совсем та же история.

— Извини, Чип, — спохватился Сережа, — рассказывай дальше.

«— А, это ты, малявка, — загрохотал великан, — зачем пожаловал? Тебя съесть — только аппетит раздразнить.

— Будь добр, дай мне пару килобайт оперативной памяти вылечить братца. Для тебя это мелочь, а ему без них один путь — на свалку.

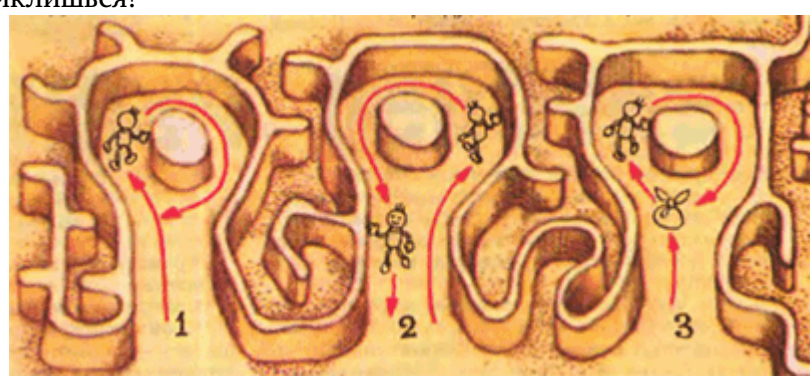
— Ха-ха-ха, мелюзга, да если я тебе даже дам пару килобайт, все равно из лабиринта не выберешься. Я и сам не знаю, как отсюда выйти.

— А вот и выйду, — засмеялся Мальчик с пальчик. — Я знаю алгоритм выхода из любого конечного плоского лабиринта. Раз я сюда пришел, то хоть один выход из твоего лабиринта есть, так что можешь обо мне не беспокоиться, дорогу я найду.

— Ну-ка, расскажи свой алгоритм, — заинтересовался Гигабайт, — а потом прогуляемся вместе наружу. Если выберемся — подарю тебе не два килобайта, а целых десять.

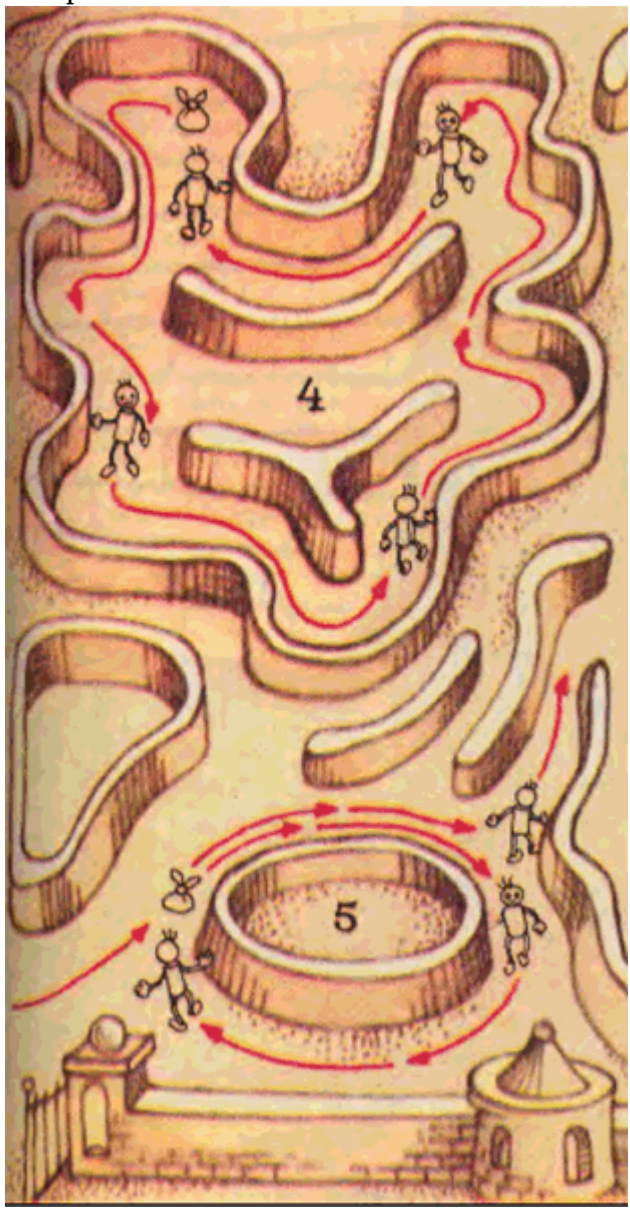
— Алгоритм очень прост, называется «Правило правой руки». Смотри: я кладу мой узелок и иду вперед вдоль правой стены. Как только встречаю развилку, поворачиваю направо. При таком алгоритме я не могу зациклиться — начать ходить по кругу, потому что, если бы я начал ходить по кругу, это бы говорило о том, что я выбирал все время не самый правый коридор. — И Мальчик с пальчик начертил на песке такой рисунок (рис. 1). — Поэтому правильный обход дал бы следующий результат (рис. 2)».

— Постой, — опять не выдержал Сережа, — а если сразу справа от тебя вот такой замкнутый коридор (рис. 3), то ты зациклишься!



— Ты невнимательно слушал. Мальчик с пальчик ясно сказал: «Кладу мой узелок». Ведь если у лабиринта нет выхода, алгоритм тоже не приведет к успеху, поэтому он и уточнил: «Раз я сюда пришел, то хоть один выход есть» (рис. 4). Поэтому, наткнувшись на свой узелок, он поступит следующим образом: возьмет его, у ближайшей развилки выберет не самый правый, а второй справа коридор (рис. 5), ведь из развилки может быть много выходов, и повторит тот же алгоритм: положит узелок, пойдет вдоль правой стенки и так далее. Можно доказать, что если выход есть, то мы

его обязательно найдем. А теперь попробуй написать программу выхода из лабиринта.



Вот что после некоторого раздумья написал Сережа:

```
ПРОГРАММА: «НАЙТИ ВЫХОД».
1. Возьмись правой рукой за стену.
2. Положи узелок у этой стены.
3. Иди по правилу «правой руки» (ситуация).
4. ЕСЛИ ситуация-выход, ТО КОНЕЦ.
ИНАЧЕ:
```

```
((5. Возьми узелок.
6. Иди вдоль правой стены до первой развилки.
7. Войди во второй справа коридор.
8. Вернись ко второму пункту программы.))

ПОДПРОГРАММА «ИДИ ПО ПРАВИЛУ» ПРАВОЙ РУКИ» (ситуация).
1. Иди вдоль правой стены до развилки.
2. ЕСЛИ увидел узелок у правой стены, ТО
((ситуация-цикл
ВОЗВРАТ)).
3. Выбери самый правый коридор.
4. ЕСЛИ выход, ТО
((ситуация-выход
ВОЗВРАТ)).
5. Выполняй первый пункт подпрограммы.
```

— Хорошо, — сказал Чип, — а теперь давай попробуем применить твою программу. Вот в журнале «Веселые картинки» напечатан лабиринт. Давай поищем путь от крота к мышке.

Вот по какому пути шли Сережа с Чипом. Если смотреть сверху, то путь может показаться не самым коротким: зачем обходить явные тупики и заворачивать во все закоулки? Но зато этот путь они прошли, как бы находясь в лабиринте, двигаясь ощупью в полной темноте, и все-таки выбрались наружу.

ОТ РЕДАКЦИИ:

Ребята! Проверьте Сережину программу на других лабиринтах. А если лабиринтов под рукой не найдется, придумайте их сами и пришлите нам. Любителям трудных задач мы предлагаем сочинить программу, которая сама будет строить лабиринты. На конверте напишите девиз: «ЛАБИРИНТ».

Секретное послание

Сережа с Чипом долго спорили, подводя итоги Большого конкурса 1987 года. Они отобрали семь лучших работ, и в каждой были интересные идеи, но, увы, ни в одной не было такого блеска, как в работах победителей прошлого года. Да и самих писем получено меньше. Тогда две с половиной тысячи, а сейчас 615. Конечно, сказки и шутки программировать легче, чем серьезные задачи.

— Знаешь, Чип, обиднее всего, что так много ребят прислали чужие программы, вместо того чтобы придумать свои.

— Да, стыдно! — Чип усмехнулся и добавил: — А ведь пройдет лет пять — и появятся компьютеры пятого поколения, которые смогут сами писать программы. Эти ребята рискуют остаться без работы в двадцать первом веке.

— Ну вот смотри, Чип. Витя Лавренко из г. Истры придумал свою игру «рыбная ловля». Идея, по-моему, интересная. Можно выбрать наживку, удочку, место, где лучше клюет, рыба может и оборвать леску, ее видно сквозь воду, так что игра азартная.

— А мне больше нравится игра «автогонки» Вити Урнышева из Петропавловска-Камчатского. У него гонщик перед началом гонки сам собирает автомобиль. Сам выбирает мотор, кузов, колеса... Интересные работы прислали Миша Погонин из Архангельска, Саша Бауров из Новосибирска, Карапет Овивян из Москвы, Петя Сурдяев из Одессы и Паша Юркин из с. Громово.

— Нет, Чип, так мы не выберем победителя. Слушай, а может, возьмем самого среднего — это же тоже одно из условий конкурса?

— Давай смотреть: средний участник конкурса-87 родился 17 августа 1975 года в 5 часов 30 минут. Ближе всех к этому моменту Олеся Матвеева из Армавира, она родилась в этот день, но на час позже. Правда, в остальном ее работа намного слабее семи лучших.

— Опять несправедливо, — возразил Сережа, — все-таки у двух Вить работы интереснее. Знаешь, а что если дать им, семи финалистам, еще одно задание. Кто лучше справится, тот и победит. Чип, помоги придумать такую задачу!

— Ну что же, мы с тобой еще не играли в шифры.

— Что, загадывать или отгадывать? — загорелись глаза у Сережи.

— И то, и другое, но только сначала я расскажу тебе одну детективную историю.

«Штирлиц взглянул на часы. Было 12.33, одна минута до

условленного времени. Он не отрываясь смотрел в окно вагона, привычно фиксируя отражения. Стрелка солидных швейцарских часов неторопливо прошла еще круг. Он вздохнул, вынул часы из жилетного кармана и переложил в правый карман ватерпруфа. Это был сигнал.

Движение нежной маленькой руки оказалось небрежно, почти незаметно. «Хорошо их учат в этой пресловутой школе», — усмехнулся про себя Штирлиц. Из осторожности он проехал еще одну остановку. Неподалеку от Генерального штаба он присел на скамейку, подождал, пока мимо проковыляла седая дама с ризеншнауцером, и только тогда вынул из кармана записку.

Судя по кириллице и размеру слов, текст был на русском языке, но шифр — новый. «Пожалуй, часа два у меня есть, раньше они не хватятся». Штирлиц любил разгадывать шифры. Частотную таблицу русских букв он помнил наизусть. Оставалось составить такую же таблицу для неизвестного текста и сравнить их. Сообщение получалось довольно странное.

О мядр щзщшпдкф юяфяспд.
— Люп впшпыкю?
— Нфпд.
— Пюлоьз.
— Пю шяыпфеьз.
— Тюп шзм дзьп?
— Чплпфзьз.
— Ьфр лпвп?
— Ьфр нэдз мпявп.
— З мдпвп фк ъыкнфзюх?
— Ьз ъоьпш пюзл ърюх
КФК чянюх: Упфхчя ямо дя наянюх: Пд о мядр ябя мзфядхлкг.

Увлекшись, Штирлиц забыл обо всем. В ключе не хватало всего трех букв, когда знакомый голос насмешливо произнес: «Игра окончена, полковник. Разведчик должен уметь проигрывать».

— Дениска, так нечестно! Мы же договорились до трех часов, а сейчас 2.30. Мне же совсем немного осталось!

— Эх, дедушка, как же ты не заметил, что я тебе в метро часы на полчаса назад подвел. А еще бывший разведчик! Ну ладно, ничья, а теперь пошли скорее, я обещал, что мы еще хлеб по дороге купим...».

— Так это что, не настоящий Штирлиц?! — возмутился Сережа. — А я-то уши развесил.

— Почему не настоящий, самый настоящий, только он уже на пенсии. Разве не может бывший разведчик поиграть после школы с внуком?

— Ну, допустим, а что за частотная таблица — я ничего не понял.

— Частотная таблица — один из самых важных инструментов шифровальщика. Возьми любую книгу и посчитай, сколько раз на какой-нибудь странице встречается буква «а», потом на другой странице, на третьей. Ты получишь примерно одинаковые числа. То же самое надо проделать с другими буквами алфавита. Можно посчитать, что любой русский текст состоит на 9,4% из букв «о», на 9% из «а» и так далее. Теперь возьми секретное послание. Мы предполагаем, что оно написано по-русски, перетасованным алфавитом. Посчитаем, какие буквы в нем встречаются и с какой частотой, сравним таблички и....

— Понял! — обрадовался Сережа.

— Только не думай, что все так просто, это ведь статистика, так можно угадать только самые часто встречающиеся буквы, так что тебе придется попотеть. А если остались еще какие-то неясности с частотной таблицей, посмотри рассказ Эдгара По «Золотой жук».

ОТ РЕДАКЦИИ.

Эту задачу мы предложили в письмах семи финалистам конкурса Чипа. Лучше всех справились Саша БАУРОВ и Карапет ОВИВЯН, они и получают калькуляторы фирм «Кассио» и «Электроника» на солнечных батарейках. Победитель шуточного конкурса Олеся МАТВЕЕВА — годовую подписку на «Пионер» 1989 года. А остальные пять претендентов на призовые места — грамоты журнала «Пионер».

Рекурсивный крокет

— Знаешь, Чип, ребята жалуются, что в последнее время наши игры стали скучнее. То ли дело, говорят, поющие поросята или 512 невест — было и смешно, и интересно. Что нам делать?

— А что тут поделаешь! Наверное, ребята правы: любая игра рано или поздно наскучит. Вот я скоро поеду путешествовать — тут уж будет о чем рассказать.

— Ну давай все-таки поиграем, ну хоть в крокет. Кстати вот уж где алгоритма не надо: гоняй себе шар по площадке, пока все ворота не пройдешь, только знай не промахнись.

Конечно, Сережа нарочно дразнил Чипа — ему очень нравилось, когда тот входил в азарт. И Чип попался на удочку.

— Это говорит программист?! Да ты что, не знаешь, что вся наша жизнь состоит из алгоритмов, не только твой дурацкий крокет? А что касается крокета, это частный случай знаменитой проблемы коммивояжера: как выбрать кратчайший маршрут через заданные точки. Для коммивояжера (бродячего торговца) это города на карте, для крокетиста — ворота на площадке.

— Ну и как выбрать этот маршрут?

— Самый короткий маршрут очень сложно выбирать, если я начну объяснять, мы с тобой последних читателей растеряем. Есть простой алгоритм выбора достаточно короткого маршрута без повторений и самопересечений. Уж так вышло, что этот алгоритм в стихах. Слушай:

Пройди по крокетной площадке AB
По правилам этим простым:
Одни лишь ворота попались тебе?
От «A» ты отправишься к ним.
«B» — угол напротив, туда ты спешишь,
Ворота пройдя без помех,
И катится шарик проворный, как мышь.
И близок желанный успех.
А **ЕСЛИ** попалось побольше ворот,
ТО все ж головы не теряй,
Не стой, удивленно разинувши рот,
Площадку на три разделяй.
По длинной, конечно, дели стороне,
Пусть поровну будет ворот,
И тот, кто рекурсию знает вполне,
Зигзагом три части пройдет.
Сначала пройди по площадке AD,

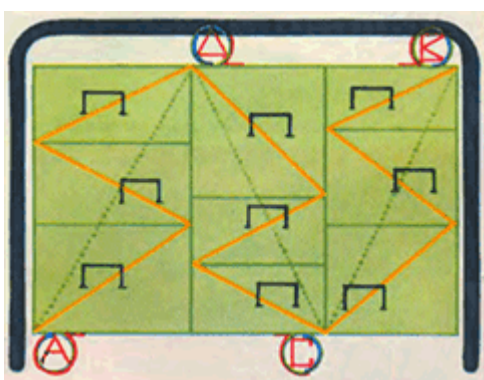
Потом по площадке DC.
СВ ты пройди, не запнувшись нигде,
И колышек стукни в конце.
Площадку прошел — ты доволен и рад,
В конце подпрограммы поставишь
ВОЗВРАТ.

— Ну как? — спросил Чип, как всегда, гордясь своим литературным упражнением.

— Да не очень... То есть стихи мне понравились, — спохватился Сережа, — только непонятно, что делать, когда будет много ворот. Вот когда одни ворота, тут все ясно: пройди их и катись в противоположный угол. Ну, когда трое ворот, тоже просто — дели площадку на три и по очереди проходи каждую...

— А ты понял, как именно проходить каждую из трех площадок? Ведь у каждой площадки есть по две диагонали, и мы их выбираем так, чтобы вместе получился зигзаг ADCB. Иначе пришлось бы делать лишнюю работу — перекатывать шар впустую из угла в угол.

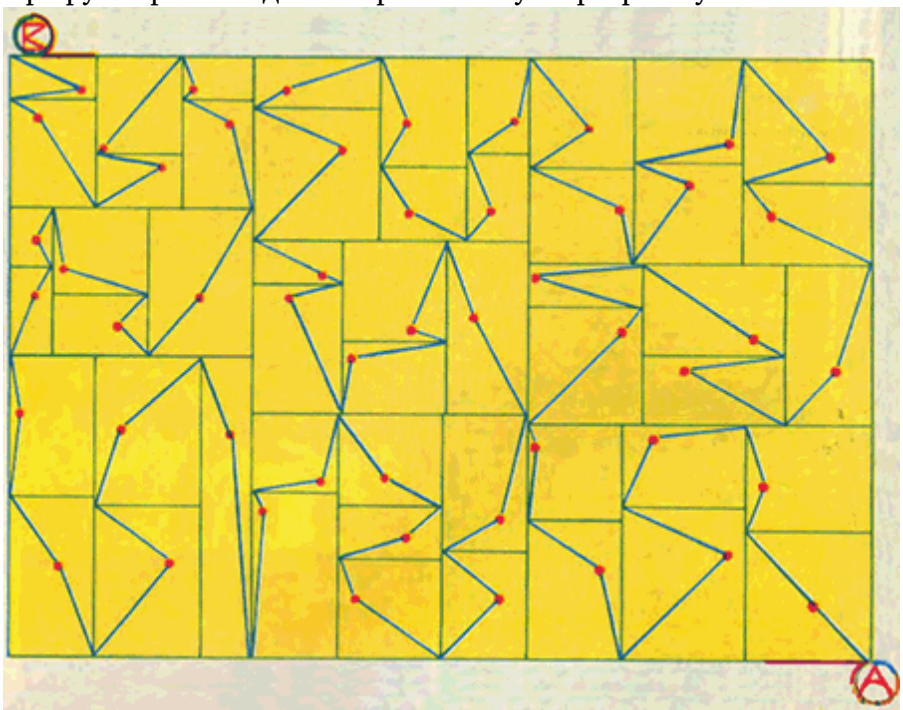
— Ну, а если будет 9 ворот, тогда я, кажется, тоже понимаю, — подхватил Сережа. — Делю всю площадку на три по трое ворот и поочередно прохожу каждую своим маленьким зигзагом. А вместе получается большой зигзаг. Вот смотри, я его нарисовал. Ага, вот почему ты указываешь два угла: начальный и конечный — чтобы проходить площадку зигзагом, друг за другом: от A к D, от D к C, от C к B. А что ты будешь делать, если число ворот не делится на три?



— Тогда можно оставить в двух площадках поровну ворот, а в третьей — на одно или на два меньше Конечно, в конце концов мы дойдем до пустых площадок, но их, я полагаю, сможет пройти любой крокетист и без моих подсказок, пусть только идет по нужной диагонали. Но вообще-то ты прав — это упущение в программе, надо было написать про пустые площадки. Пусть

ребята это условие впишут в нашу стихотворную программу.

Сереже очень понравился алгоритм рекурсивного крокета, и он его в своем кружке запрограммировал для настоящего компьютера. Посмотрите, как нарисовал компьютер крокетный маршрут через пятьдесят ворот. А какую программу напишете вы?



Чип и Сережа ждут, что вы пришлете свои программы и на конверте поставите девиз **«Крокет»**.

ОТ РЕДАКЦИИ:

На время Чип расстается с нами, ребята. Он едет в США работать над советско-американским проектом «Фобос». Обещал Сереже писать письма. Когда Сережа получит эти письма, мы их напечатаем.

В ожидании Чипа

Об электронных переводчиках и лучших портных Кливленда

Сколько бы теперь ни набирал Сережа на калькуляторе 1234 + 5678, Чип больше не появлялся. Далеко был Сережин дружок, в Америке, в городе Сан-Диего. Он улетел в США работать над советско-американским космическим проектом «Фобос». Сережа уж весь атлас исползал, а нашел Сан-Диего. Оказывается, он в штате Калифорния, почти на самой границе с Мексикой и в то же время на берегу Тихого океана.

«Ах, Чип, Чип. Что же ты не пишешь? А ведь обещал. Я так привык к тебе», — думал Сережа.

Забросив атлас, Сережа пошел в папин кабинет посмотреть, нет ли чего почитать про Чипову жизнь и жизнь его братии. Вообще-то шарить на папином столе Сереже не разрешалось. Но сегодня он нарушил запрет и, аккуратно приподняв бумаги, исписанные быстрым папиным почерком, вытянул несколько номеров журнала «В мире персональных компьютеров».

Усевшись в кресло, Сережа погрузился в чтение. Текст был научный, сложный, но кое-что интересное Сережа все-таки вычитал.

Оказывается, в Англии разрабатывается устройство автоматического перевода речи с одного языка на другой. Установлено оно будет на телефонных линиях. Англичанин сможет понять японца, хотя каждый из них будет говорить на родном языке. Как работает устройство? К телефону подключается персональная ЭВМ, на экране дисплея появляется текст произносимой фразы. «Переводчик» знает 1000 слов на английском, французском, испанском, японском языках.

Две японские фирмы изобрели щетки, которые моют самолет всего лишь за час. Управляют щетками компьютеры, а обслуживают эту установку, размещенную в одном из аэропортов Токио, всего пять человек.

Вычислительные машины умеют не только решать научные задачи, но и могут сочинять стихи. Вот какие стихи написала машина по программе двух английских студентов:

Камень стоит шелковый,
как сильный шелк остается

ярким в шелково-зеленой
солнечной ночи.
Огонь становится шелковым,
в то время как надежда ласкает
как неясное неясный снег.

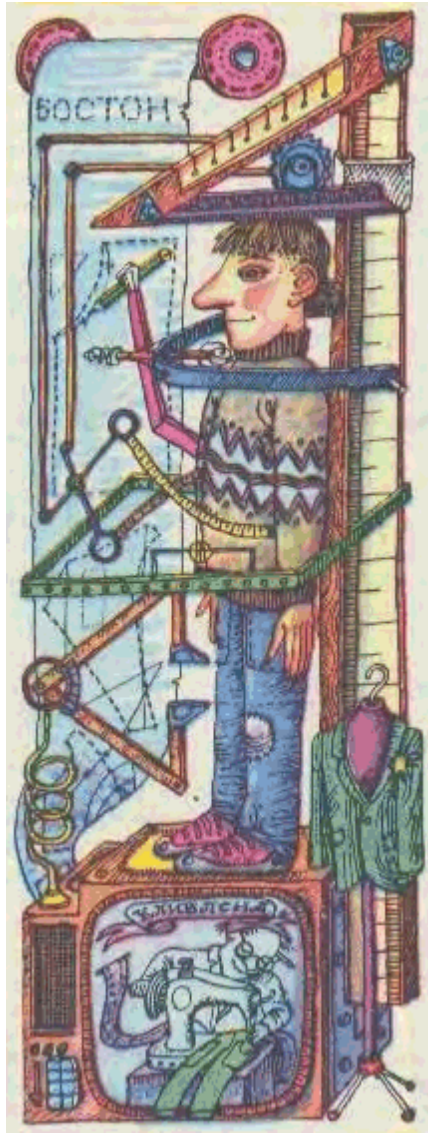
А в Советском Союзе создана программа, сочиняющая волшебные сказки.

Одна французская фирма разработала вычислительную машину, которая откликается на человеческую речь. Машина понимает 44 команды и может открыть дверной замок, набрать телефонный номер, который вы ей продиктуете, включить в комнате свет...

В универсальном магазине бостонской фирмы «Альберт Эндрюс» с помощью компьютеров шьют на заказ мужские костюмы. Покупатель выбирает фасон и ткань, после чего специальное устройство с большой точностью снимает с него мерки. Персональный компьютер передает эти мерки в Кливленд, где находится пошивочная мастерская. Здесь ЭВМ составляет рисунок раскроя, затем другое устройство кроит ткань. Правда, шьют костюм люди — профессиональные портные и швеи. При такой работе стоимость костюма снижается в два раза.

Сережа посмотрел по карте США, где Бостон, а где Кливленд. Ничего себе расстояние! Наверное, в Кливленде живут самые лучшие портные, раз бостонцы хотят шить костюмы только у них. Но как костюмы из Кливленда пересылаются обратно в Бостон?

Додумать этот вопрос Сережа не успел, потому что в прихожей хлопнула дверь и зажегся свет. Это бабушка вернулась. Пришлось быстро засунуть журналы под папины бумаги и сесть за уроки.


БОСТОН

Работа над ошибками

На этот раз Сережа не стал тайком забираться в папин кабинет, а, описав несколько кругов вокруг отца, умиротворенно листающего газеты после дневных трудов, спросил его напрямик:

— Пап, а в твоих научных журналах пишут что-нибудь интересное про компьютеры? Ну, вот в «Science News», например?

— А как же, обязательно пишут.

— Может, ты мне почитаешь оттуда? Только по-русски.

— Знаешь что, — вдруг оживился папа, — давай-ка попробуем почитать вместе. Все незнакомые тебе слова я подскажу. А что касается Present Perfect или Past Continous — тут уж изволь соображать сам. Программиста, не знающего английского языка, я не могу представить.

Они пододвинули к письменному столу еще одно кресло, взяли журнал «Popular Science», название которого они перевели как «Наука для всех», англорусский словарь Мюллера и начали вычитывать занимательные истории про чипову жизнь.

— Давай прочтем хотя бы вот это, — сказал папа и ткнул пальцем в колонку справа.

Победителем конкурса, организованного японской фирмой «Сони», на лучший бытовой телевизор будущего стал студент колледжа. Он сконструировал телевизор в виде двуногого робота. Робот движется за хозяином из комнаты в комнату, исполняет танцы под музыку и может принимать различные позы.

ЭВМ используется в США для быстрого поиска и установления личности по отпечаткам пальцев. Машина может автоматически восстановить некачественно снятые или смазанные отпечатки, устанавливает личность и по неполным отпечаткам пальцев. За один год такая система установила личности 1200 подозреваемых, в том числе 90 человек, разыскиваемых по делам, связанным с убийствами.

В ФРГ создана автоматическая система резервирования мест в гостиницах. ЭВМ понимает человеческую речь и работает как портье, давая информацию об обстановке комнат, ценах и так далее. Например, может происходить такой диалог. Клиент спрашивает: «Есть ли у вас большая комната?» «Да», — отвечает ЭВМ. Клиент уточняет: «Есть ли в ней большое окно?» «Да», — отвечает ЭВМ. «Она тихая?» В ответ ЭВМ говорит: «Да, вполне тихая».

Для испытания военного обмундирования в боевых условиях в США создан человекоподобный робот. Скелет робота изготовлен из трубок и шарниров и покрыт искусственной кожей, которая может регулировать температуру. Робот «дышит», увеличивая и сокращая объем грудной клетки.

Вычислительные машины уже помогают врачам ставить диагнозы, назначать процедуры и лекарства. Но, оказывается, полностью доверять машинам рано. В США за последние пять лет количество ошибочных диагнозов машин-докторов возросло в два раза. Так, из-за ошибки в программе, которую не обнаружили во время испытаний, нескольким людям была назначена неправильная доза облучения, что привело к тяжелым последствиям.

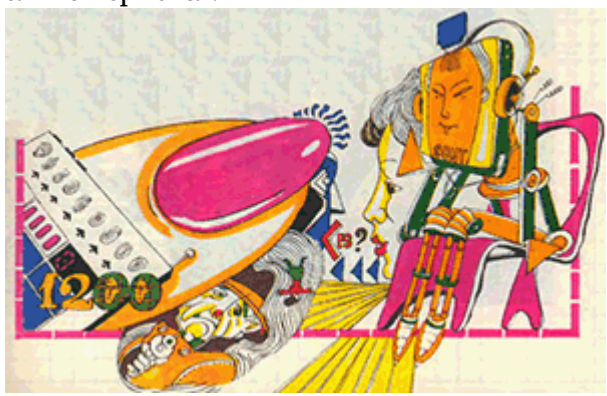
— Па, а компьютеры часто ошибаются? — прочитав последнее известие, заинтересовался Сережа.

— Случается.

— А давай, я буду собирать сообщения об их ошибках. Может, кто-нибудь потом изучит все эти факты и придумает, как избежать ошибок?

И так этот случай ошибки компьютера открыл Сережину коллекцию.

А Чипа Сережа решил ждать хоть сто лет. Ведь, если ждешь, друг обязательно вернется.



notes

Примечания

1

Дорогие ребята, не думайте, что Чип вас обманул. Чипом действительно называется самая главная деталь — «мозг» калькулятора.

2

Ребята, эту сказку вы не найдете ни в одном сборнике сказок. Ее Чип сочинил сам, как сумел.

3

Тут Сережа ошибся. ЖЭК — жилищно-эксплуатационная контора — не строит, а эксплуатирует дома.

4

Подпрограмма — это кусочек программы, который можно вставлять в любое место. При этом можно что-то менять в этом кусочке, надо только точно указать, что и как. В этом отношении подпрограмма напоминает платье-полуфабрикат, которое подгоняют по фигуре. Подпрограмму можно вставлять и в другую подпрограмму, а ту — еще в одну, как матрешки. Если вам все равно непонятно, не смущайтесь, прочтите дальше, поиграйте с этим стишком, и вам постепенно все станет ясно.

5

Ребята, Чип действительно может сделать такое чудо. Многие из вас видели электронные игры-приставки к телевизору. Этими играми управляют такие же маленькие чипы, что и сидят в компьютерах.

6

Джинны и ифриты — добрые или злые духи в арабском фольклоре.

7

БЛОК - это любой набор команд, а двойные скобки нужны, чтобы видеть, где он начинается, а где кончается.

comments

Комментарии

1

(прим. ред.) По всей видимости, речь идёт о Норберте Винере - американском учёном, выдающемся математике и философе, основоположнике кибернетики и теории искусственного интеллекта.

2

(прим. ред.) Возможно, Чип имеет в виду случай, произошедший с космическим аппаратом Mariner 1, во время запуска которого 22 июля 1962 была потеряна связь с наводящей системой на Земле, в результате чего управление взял на себя бортовой компьютер, программа которого содержала ошибку, названную позднее "самый дорогой дефис в истории".